Computer Applications

Solution (ICSE)

Frank
EMU BOOKS

Chapter 1
Revision Tour I

Mental Drill

A. Tick (✓) the correct option.

1. Name the programming technique that implements programs as an organised collection of interactive objects.

   a. Procedure Oriented Programming
   b. Modular Programming
   c. Object Oriented Programming
   d. None of these

   Ans. c. Object Oriented Programming

2. Name the characteristics of Object Oriented Programming that hides the complexity and provides a simple interface.

   a. Encapsulation
   b. Polymorphism
   c. Abstraction
   d. Inheritance

   Ans. c. Abstraction

3. Which among the following operator is used to access individual members of an object?

   a. . (dot)
   b. + (plus)
   c. – (minus)
   d. / (divide)

   Ans. a. . (dot)

4. Which among the following modifier is used in a 'class'?

   a. public
   b. default
   c. Both a and b
   d. None of these

   Ans. c. Both a and b

5. Which among the following is a valid class name?

   a. Simple Interest
   b. SimpleInterest
   c. 1SimpleInterest
   d. Simple@Interest

   Ans. b. SimpleInterest

6. Which among the following is a valid object name?

   a. obj1
   b. 1obj
   c. Obj 1
   d. Obj#1

   Ans. a. obj1

7. Which among the following is a valid float literal?

a. 12.36f b. 12.36F

c. 12.36 d. Both a and b

Ans. d. Both a and b

8. Which among the following is a valid octal integer literal?

a. 0178 b. 675

c. 0675 d. 0X675

Ans. c. 0675

9. Which among the following is a valid method of initialising?

a. boolean f=true; b. boolean f=True;

c. boolean f='true'; d. None of these

Ans. a. boolean f=true;

10. Which among the following is not a punctuator?

a. ; semicolon b. , comma

c. : colon d. . dot

Ans. d. . dot

11. Which among the following is not a primitive data type?

a. int b. float

c. String d. char

Ans. c. String

12. What is the largest possible value that can be stored in short data type?

a. 215^{-1} b. 231^{-1}

c. 27^{-1} d. 263^{-1}

Ans. a. 215^{-1}

13. If a is of type int and b is of type float what would be the resultant data type of a+b?

a. int b. float

c. double d. short

Ans. b. float

B. State whether the following statements are True (T) or False (F).

1. Encapsulation refers to the art of hiding the complexities and giving a simple interface. **F**
2. Procedure Oriented Language follows top down approach. **T**
3. Java is an example of Object Oriented Language. **T**

4. Hiding the complexity and giving a simple interface is called Inheritance. **F**
5. Abstraction is same as encapsulation. **F**
6. An object is called a class factory. **F**
7. A class is an instance of an object. **F**
8. A class is a mechanism to implement encapsulation. **T**
9. Data members in a class is used to represent the characteristic of an object. **T**
10. The new operator is used to create an object. **T**
11. It's a rule to have a class-name beginning in capital letter. **F**

C. Fill in the blanks.

1. A **class** is a template that binds together data and methods together.
2. The values in the attributes of an object is called the **state** of an object.
3. The **dot** operator is used to access the individual members of a class.
4. The keyword **new** is used to allocate memory space for an object.
5. The default and **public** access modifier is used with a class.
6. An object is an identity with certain **characteristic** and **behaviour**.
7. The values/attributes of the characteristics of an object are called the **state** of an object.
8. All the complexities of a program should be **encapsulated** in such a way so that **abstraction** is obtained.
9. **Inheritance** allows us to encompass the parent class' state and behaviours into its child..
10. Poly-means **many** and Morphism means **forms**.

Section A

Answer the following questions:

1. How are objects implemented in software?

Ans. In a software the characteristics of an object are represented through data members and behaviour is represented through member functions.

2. What is abstraction? How is encapsulation related to it?

Ans. Abstraction is a principle of Object Oriented Programming (OOP) that hide certain details and only show the essential features of the object.

3. Define Encapsulation.

Ans. Encapsulation is a principle of Object Oriented Programming (OOP) that binds together characteristics and behaviour of an object into a single unit represented by a class.

4. What is Inheritance?

Ans. Inheritance is the concept that when a class of objects is defined, any subclass that is defined can inherit the definitions of one or more general classes.

5. What is Object Oriented Programming?

Ans. Object Oriented Programming *(or OOP) is a technique of implementing programs which are organized as a co-interactive collection of objects, each of which represents an instance of a class.*

6. State three differences between Procedure Oriented Language and Object Oriented Languages.

Ans.

Procedure Oriented Programming	Object Oriented Programming
A large program is divided into smaller segments or procedures.	A program is represented as an object.
More importance is given to the program rather than the data.	More importance is given to the data rather than the program.
It follows top down approach.	It follows bottom up approach.

7. State the four characteristics/principles of Object Oriented Programming.

Ans. Encapsulation, Abstraction, Polymorphism and Inheritance.

8. What are keywords? Give an example.

Ans. A keyword is a reserved word that has a special significance to the compiler and cannot be used anywhere else other than what it is intended for. Example- for, if, else, while etc.

9. What are identifiers?

Ans. Identifiers are the names of variables, methods, classes, packages and interfaces.

10. What is a literal?

Ans. Literal is a constant value that can be assigned to a variable.

11. Why is an object called an instance of a class?

Ans. An object is called an instance of a class as every object created from a class gets its own instances of the variables defined in the class. Multiple objects can be created from the same class.

12. Write one difference between primitive data type and composite data type.

Ans.

Fundamental data type	Composite data type
These are inbuilt data type provided by the Java Language.	These are data types created by the user using fundamental or user defined data type or both.

The size of it is fixed.	The size of different user-defined data type depends upon the size of the individual components of it.
These data types are available in all parts of a program within a class.	These data types are available only as specified by the access specifiers.

13. Give one example each of primitive data type and composite data type.

Ans. Primitive data type: int, short, boolean, char etc.
Composite data type: class, arrays, interface etc.

14. State two differences between a class and an object.

Ans.

Object	Class
Object is an instance of a class.	Class is a blueprint or template from which objects are created.
Object is a real world entity such as pen, laptop, mobile, bed, keyboard, mouse, chair etc.	Class is a group of similar objects.
Object is a physical entity.	Class is a logical entity.
Object is created through new keyword mainly e.g. Student s1=new Student();	Class is declared using class keyword e.g. class Student{}
Object is created many times as per requirement.	Class is declared once.
Object allocates memory when it is created.	Class doesn't allocated memory when it is created.

15. Give one point of difference between unary and binary operators.

Ans. Unary operator works on a single operand and Binary operator works on 2 operands.

16. What do you understand by type conversion?

Ans. Type conversion is the process of converting one Primitive data type to another primitive data type. It may be done either implicitly or explicitly.

17. State the difference between a Boolean literal and a character literal.

Ans. A boolean literal consist of only two values i.e. true or false. A character literal on the other hand is any character enclosed within single quotes.

18. Identify the literals as given below:

i. 0.5 ii. 'A' iii. false iv. "a"

Ans. i. double ii. char iii. boolean iv. String

19. Which two access specifier is used in a class declaration?

Ans. public and default

20. Why is a class called an object factory?

Ans. A class is called an object factory because objects are created from a class. An object is an instance of a class.

21. Evaluate the value of n if the value of p=5 and q=19:

int n = (q-p)>(p-q)?(q-p):(p-q);

Ans. n=14

22. What is meant by precedence of operators?

Ans. When several operations occur in an expression, each part is evaluated and resolved in a predetermined order called operator precedence.

23. What is Operator Associativity?

Ans. Operator associativity of an operator is a property that determines how operators of the same precedence are grouped in the absence of parentheses; i.e. in what order each operator is evaluated when two operators of same precedence appear in an expression.

24. State the difference between an accumulator and counter.

Ans. Accumulator is a variable that is used to add or accumulate a list of items. Counter on the other hand is a variable, which is used to keep track of the number of times an operation is being performed.

25. What does a class encapsulate?

Ans. A class encapsulates characteristics represented by data member and behaviour represented by member methods.

26. State the Java concept that is implemented through:

i. A super class and a sub-class.

ii. The act of representing essential features without including background details.

Ans. i. Inheritance

ii. Abstraction

27. Write a statement in Java that will declare an object named si of the SimpleInterest class.

Ans. SimpleInterest si = new SimpleInterest();

28. Rewrite the following program after removing the errors, underlining each correction:

```
class My Class
        {
            int a, b;
            void initialize( )
            {
```

```
        a=5;
        b=6;
    }
    void show ( )
    {
    System.out.println (a+ ‘’ ‘’ + b);
    }
    static void main( )
    {
    My Class ob = new My Class ( );
    ob. initialize ( );
    show ( ). ob;
    }
}
```

Ans.

```
class MyClass
{
        int a, b;
        void initialize( )
        {
        a=5;
        b=6;
        }

void show ( )
{
        System.out.println (a+ ‘’ ‘’ + b);
}

static void main( )
        {
        MyClass ob = new MyClass( );
        ob. initialize( );
        ob.show ( );
        }
}
```

29. Which among the following are invalid class names in Java? State with reasons.

i. Compound Interest ii. 1MyClass iii. MyClass$ iv. Myclass#

v. My@Class

Ans. i. Invalid, as a variable name cannot have a blank space.

ii. Invalid, as a variable name cannot begin with a digit.

iv. Invalid, as a variable name cannot have a special character, like #.

v. Invalid, as a variable name cannot have a special character, like @.

30. What is the resultant data type of the following mathematical expression?

a+b*c-d

a. where a is int, b is int, c is float and d is float type

b. where a is float, b is long, c and d is of int type

c. where a is of double and b,c and d are of int type

d. where a is char and b,c and d are of int type

e. where a, b, c and d are of int type, however the expression is slightly modified as (a+b*c-d)/7.0

Ans. a. float

b. float

c. double

d int

e. double

31. What will be the output of the following programs?

a.

```
class Output1
{
    static void main()
    {
        int a=5,b=6,c;
        c=a+b%2;
        b=a+2*c;
        System.out.println(c);
        System.out.println(b);
    }
}
```

b.

```
class Output2
{
    static void main()
    {
        int a=12,b=13,c=6;
        c+=a+b%2;
        b-=a+2*c;
        System.out.println(c);
        System.out.println(b);
    }
}
```

c.

```
class Output3
{
    static void main()
    {
        int a=5;
        a++;
        System.out.println(a);
        a-=(a--) - (--a);
        System.out.println(a);
    }
}
```

[ICSE 2014]

d.

```
class Output4
{
    static void main()
    {
        int a=12,b=13,c;
        c=a++ + ++a/7 + b++;
        System.out.println(a+"\t"+b+"\t"+c);
        c=++a + a++/3 + ++b;
        System.out.println(a+"\t"+b+"\t"+c);
    }
}
```

e.

```
class Output5
{
    static void main()
    {
        int a=10,b=5;
        boolean f1,f2;
        f1=a>=b;
        b+=9;
        f2=a<=b;
        System.out.println(f1+"\t"+f2);
        System.out.println(a+"\t"+b);
    }
}
```

f.

```
class Output6
{
    static void main()
    {
        int a=1,b=2,c=3;
        boolean f1,f2;
        f1=a<=b++;
        f2=c+9>b++;
        System.out.println(f1+" "+f2);
        System.out.println(a+b+c);
    }
}
```

g.

```
class Output7
{
    static void main()
    {
        int a=12,b=5,c=8;
        boolean f1,f2;
        f1=a>b && b>c;
        f2=a>b || b>c;
        System.out.println(f1+" "+f2);
    }
}
```

h.

```
class Output8
{
    static void main()
    {
        int a=10,b=16,c;
        boolean f;
        c=(a>b)?++a:++b;
        System.out.println(a+" "+b+" "+c);
        f=a%2==0;
        System.out.println(f);
    }
}
```

Ans. a. 5
15

b. 19
-37

c. 6
4

d. 14 14 27
16 15 35

e. true true
10 14

f. true true
8

g. false true

h. 10 17 17
true

32. What will be the output when the following code segment is executed?

System.out.println(“The king said \”Begin at the beginning!\ “to me”);

Ans. The king said “Begin at the beginning!” to me

33. System.out.print(“BEST “);

System.out.println(“OF LUCK”);

Choose the correct option for the output of the above statements:

i. BEST OF LUCK

ii. BEST
OF LUCK

Ans. i. BEST OF LUCK

34. What is the value of y after evaluating the expression given below?

y+=++y + y-- + --y; when int y=8

Ans. y=33

35. Give the output of the following expression:

a+=a++ + ++a + --a + a--; when a=7.

Ans. a=39

36. If int y=10 then find:

int z=(++y * (y++ +5));

Ans. y=12 z=176

37. What are the values of x and y when the following statements are executed?

int a = 63, b = 36;

boolean x = (a < b) ? true : false;

int y= (a > b) ? a : b;

Ans. x=false y=63

38. What will be the result stored in x after evaluating the following expression?

int x=4;

x += (x++) + (++x) + x;

Ans. x=20

39. What will be the output of the following code?

int k = 5, j = 9;

k += k++ – ++j + k;

System.out.println("k=" +k);

System.out.println("j=" +j);

Ans. k= 6

j= 10

40. What is the result stored in x, after evaluating the following expression?

int x = 5;

x = x++ * 2 + 3 * –x;

Ans. x=-8

41. Write the output of the following:

System.out.println("Incredible"+"\n"+"world");

Ans. Incredible

World

42. Name the operators listed below:

i. < ii. ++ iii. && iv. ?:

Ans. i. Relational Operator

ii. Unary Operator (Increment)

iii. Logical operator (and)

iv. Ternary operator

43. State the number of bytes occupied by char and int data types.

Ans. char = 2 bytes

int = 4 bytes

44. Write one difference between / and % operator.

Ans. / is used to find the quotient and % is used to find the remainder, when 2 numbers are divided.

45. Name the primitive data type in Java that is:

i. a 64 bit integer and is used when you need a range of values wider than those provided by int.

ii. a single 16-bit Unicode character whose default value is '\u0000'.

Ans. i. long

ii. char

SECTION B

Programming Questions

1. Write a program to find the sum and difference between 25 and 16 using variables in different lines.

Ans.
```
class q1
{
        static void main()
        {
            int a=25,b=16,s,d;
            s=a+b;
            d=a-b;
            System.out.println("Sum="+s);
            System.out.println("Difference="+d);
        }
}
```

2. Write a program to find the product of 5, 7 and 12 using variables.

Ans.
```
class q2
{
        static void main()
{
```

```
        int a=5,b=7,c=12,d;
        d=a*b*c;
        System.out.println("Product="+d);
    }
}
```

3. Write a program to find the product of the sum and difference between 17 and 2 using variables.

Ans.

```
class q3
{
    static void main()
    {
        int a=17,b=2,s,d;
        s=a+b;
        d=a-b;
        System.out.println("Sum="+s);
        System.out.println("Difference="+d);
    }
}
```

4. Write a program to average of 36, 45 and 53 using variables of adequate data type.

Ans.

```
class q4
{
    static void main()
    {
    int a=36,b=45,c=53;
    float av;
    av=(float)(a+b+c)/3;
    System.out.println("Average="+av);
    }
}
```

5. Write a program using int variables to find the sum of three numbers say 15, 36 and 45 and subtract the result from 100 using variables.

Ans.

```
class q5
{
    static void main()
    {
        int a=15,b=36,c=45,d=100,e;
        e=d-(a+b+c);
        System.out.println("Answer="+e);
    }
}
```

6. Write a program to display the names of five fruits with a single System.out.println(); statement, but in different lines.

Ans.
```
class q6
{
        static void main()
        {
        System.out.println("Apple\nOrange\nGuava\nBanana\nLemon");
        }
}
```

7. Write a program to find the sum, difference and product of 12.35 and 7.3 using variables with a single System.out.println(); statement, however with horizontal tab space in the result.

Ans.
```
class q7
{
        static void main()
        {
                float a=12.35f,b=7.3f,s,d,p;
                s=a+b;
                d=a-b;
                p=a*b;
                System.out.println("Sum="+s);
                System.out.println("Difference="+d);
                System.out.println("Product="+p);
        }
}
```

8. Write a program using float type variables to find the area and perimeter of a square whose side is 12.5 cm.

Ans.
```
class q8
{
        static void main()
        {
                float s=12.5f,a,p;
                a=s*s;
                p=4*s;
                System.out.println("Area="+a);
                System.out.println("Perimeter="+p);
        }
}
```

9. Write a program using int variables to find the area and perimeter of a rectangle of length 12cm and breadth 8cm.

Ans.
```
class q9
 {
   static void main()
```

```
  {
    int l=12,b=8,a,p;
    a=l*b;
    p=2*(l+b);
    System.out.println("Area="+a);
    System.out.println("Perimeter="+p);

  }
}
```

10. Write a program using variables to find the profit and profit percent of a certain transaction where

S.P.= ₹10000 and C.P.= ₹ 7000.

Ans.

```
class q10
{
  static void main()
  {
    float sp=10000,cp=7000,p,pp;
    p=sp-cp;
    pp=p/cp*100;
    System.out.println("Profit="+p);
    System.out.println("Profit Percent="+pp);

  }
}
```

11. Write a program to input the Principal, Rate and Time and calculate the Simple Interest.

$$\text{Simple Interest} = \frac{(\text{Principal}*\text{Rate}*\text{Time})}{100}$$

Ans.

```
import java.util.*;
class q11
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    float p,r,t,si;
    System.out.println("Enter the principal, rate and time:");
    p=sc.nextFloat();
    r=sc.nextFloat();
    t=sc.nextFloat();
    si=(p*r*t)/100;
    System.out.println("Simple Interest="+si);
  }
}
```

12. Write a program using variables to find the area and circumference of a circle whose radius is 12cm.

Note that :

Area of a circle= πr^2

Circumference of a circle = $2 \pi r$

where r is the radius of the circle and π is 3.142.

Ans.

```
class q12
{
  static void main()
  {
    float r=12,a,pie=3.142f,c;
    a=pie*r*r;
    c=2*pie*r;
    System.out.println("Area="+a);
    System.out.println("Circumference="+c);

  }
}
```

13. Write a program to initialise two integer variables a and with 5 and 6 respectively and interchange them. Thus after interchanging, a and b will be 6 and 5 respectively.

Ans.

```
class q13
{
  static void main()
  {
    int a=5,b=6,c;
    System.out.println("Before Interchange: a="+a+"b="+b);
    c=a;
    a=b;
    b=c;
    System.out.println("After Interchange: a="+a+"b="+b);
  }
}
```

14. Write a program to initialise three int variables a, b and c with 234, 456 and 712 and store the sum of the last digits of the variables into d and display it.

Ans.

```
class q14
{
  static void main()
  {
    int a=234,b=456,c=712,d;
    d=a%10+b%10+c%10;
    System.out.println("Sum="+d);
```

```
    }
}
```

15. Write a program to initialise an int variable a with 76498 and from it extract the first digit and store it into a variable f and extract the last digit into a variable l and display both these digits.

Ans.
```
class q15
{
  static void main()
  {
    int a=76498,f,l;
    f=a/10000;
    l=a%10;
    System.out.println("First Digit="+f);
    System.out.println("Last Digit="+l);
  }
}
```

16. Write a program using ternary operator to check whether 27 is a multiple of 3 or not.

Ans.
```
class q16
{
  static void main()
  {
    System.out.println((27%3==0)?"Multiple of 3":"Not a multiple of 3");
  }
}
```

Chapter 2
Revision Tour II

Mental Drill

A. Tick (✓) the correct option.

1. Which among the following is not a valid error in Java?

   a. Syntax errors b. Logical errors

   c. Run-time errors d. Technical errors

   Ans. d. Technical errors

2. Which among the following Scanner methods allows us to input a number with a decimal point?

   a. nextInt() b. nextFloat()

   c. nextDecimal() d. nextPoint()

   Ans. b. nextFloat()

3. The output in BlueJ occurs in which window?

   a. Console window b. Terminal window

   c. Both a and b d. None of these

   Ans. b. Terminal window

4. The input in BlueJ occurs in which window?

   a. Console window b. Terminal window

   c. Both a and b d. None of these

   Ans. b. Terminal window

5. Assigning value to a variable during declaration is called.

   a. Declaration b. Assignment

   c. Initialisation d. None of these

   Ans. c. Initialisation

6. Which among the following is used to represent single-line comment?

   a. // b. /* c. \\ d. <!—

   Ans. a. //

7. Which among the following is a logical error?

   a. Missing semicolon b. Mismatched braces in classes and methods.

   c. Misspelled keywords and identifiers. d. Addition is required but subtraction is performed.

   Ans. d. Addition is required but subtraction is performed.

8. Which among the following represents a syntax error?

   a. Dividing an integer by zero.

   b. Accessing an element that is out of bounds of an array.

   c. Trying to store a value which is incompatible to a certain data-type.

   d. Missing semicolon

Ans. d. Missing semicolon

9. If the data that is to take part in a calculation in a method is fixed, which form of input is necessary?

   a. Initialisation

   b. Parameterised input

   c. Scanner input

   d. None of these

Ans. a. Initialisation

10. In case you need to give a proper message (prompt string) before taking an input from the user, which would be the most preferable method?

    a. Parameterised input

    b. Initialisation

    c. Scanner input

    d. None of these

Ans. c. Scanner input

B. Fill in the blanks.

1. The **break** statement is used to avoid fall through.
2. The **next().chart(0)** function of the Scanner class is used to accept a character from the user.
3. A loop within another loop is called **nested** loops.
4. The **relational/comparison** operator is used to compare two quantities.
5. The else block is preceded by the **if** block.
6. The **sqrt()** function of the Math class is used to return the square root of a number.
7. The return type of cbrt( ) function is **double**.
8. The **continue** jump statement is used to skip the remaining statements in a loop.
9. Single line comment and **multiline** comments are two types of comments in Java.
10. Any error in the grammar of the language is a **syntax** error.

C. State whether the following statements are True (T) or False (F).

1. Scanner class is present in the java.lang package. **F**
2. Math.abs( ) is used to find the absolute value of a number. **T**
3. The return type of Math.sqrt( ) function is float. **F**
4. The fraction 1/2 will evaluate to 0.5. **F**

5. The continue statement in a switch block is used to avoid fall through. F

6. The default statement is essential in a switch block. F

7. The for loop is an entry controlled loop. T

8. The while loop is an exit controlled loop. F

9. The do-while loop is generally used when the number of iterations is known. F

10. You can have only the for loop as the nested loops. F

SECTION A

Answer the following questions.

1. Name the functions of the Scanner that is used to:

(i) Accept a number of long data type

(ii) Accept a group of alphabets.

Ans. (i) nextLong() (ii) nextLine()

2. What is a bug? What is debugging?

Ans. An error in a program is called a bug and the process of removing it is called debugging.

3. What are the different types of errors that may occur in a Java program?

Ans. Syntax Error, Logical Error and Run-time Error.

4. What are syntax errors? Give two examples.

Ans. A syntax error is an error in the syntax of a sequence of characters or tokens that is intended to be written in a particular program.

Examples:

- Mismatched braces in classes and methods
- Wrongly accessing variable, whose visibility doesn't exist
- Misspelled keywords and identifiers

5. What are run-time errors? Give two examples.

Ans. An error that occurs during the execution of a program is called run time error.

Example:

- Dividing an integer by zero.
- Accessing an element that is out of bounds of an array.
- Trying to store a value which is incompatible to a certain datatype.

6. What are comments? Name the different types of comments used in Java.

Ans. A comment is a programmer-readable explanation or annotation in the source code of a computer program.

Types: Single-line comment and Multiline comment

7. What are conditional statements? With which commands do you implement conditional statement in Java?

Ans. Conditional construct are specific statements that allow us to check a condition and execute certain parts of code depending on whether the condition is true or false.

Implementation is done using: if, if-else, if-else if-else, switch-case and ternary operators.

8. When are braces optional or necessary in conditional statements?

Ans. Giving braces is optional for if/else in case there is only one statement to be executed.

9. What is the difference between the Scanner class functions next() and nextLine()?

Ans. next() can read the input only till the space. It can't read two words separated by space. Also, next() places the cursor in the same line after reading the input. nextLine() reads input including space between the words (that is, it reads till the end of line \n).

10. What are relational operators?

Ans. The relational operator allows you to test or define some kind of relation between two entities.

11. State the difference between = and = =.

Ans. In Java, "=" is the assignment operator and "==" is a comparison operator.

12. What are logical operators? Explain each of them with the help of an example.

Ans. The logical operators are used to combine simple relational statements into more complex expressions.

Logical Operator	Name	Behaviour
&&	AND operator	Checks whether the conditions preceding and succeeding it is true or not.
||	OR operator	Checks whether either of the conditions preceding and succeeding it is true or not.
!	NOT operator	Just negates the logic of the condition succeeding it to check for its validity.

13. State the difference between while and do while loop.

Ans. The while loop is an entry controlled loop, whereas the do-while loop is an exit controlled loop.

14. State the difference between entry controlled loop and exit controlled loop.

Ans.

Difference between entry controlled and exit controlled loop	
Entry Control Loop	**Exit Control Loop**
The test expression or condition is checked before entering the loop.	The test expression or condition is checked after entering the loop.
It the test expression evaluates to false, the loop doesn't get executed.	Even if the test expression evaluates to false, the loop gets executed atleast once.

15. Give the general syntax of a while-loop. How do you create infinite loops using a while-loop structure?

Ans. The syntax of a while loop is,

while (condition or test-expression)

{

bodyoftheloop;

}

One of the method of creating an infinite loop is to write 'true' in the test-expression of the loop.

while(true)

System.out.println("Infinite");

16. Give the general syntax of a do-while loop. How do you create infinite loops using do-while loop structure?

Ans. The syntax of the dowhile loop is:

```
do
{
    statement;
}while (condition);
Infinite loop using do-while loop:
do
 {
     System.out.println("Infinite");
  }while(true);
```

17. Compare loops according to its usage.

Ans. Generally the for-loop is used when the number of iterations is known. The while loop is generally used when the number of iterations is not known. The do-while loop is generally used when it is necessary to execute the loop at least once.

18. What are Nested-loops? Give an example of a nested for-loop to show its functionality.

Ans. A loop within a loop is called a nested loop.

Example of a nested loop is:

```
public class Loops
{
    static void Nested()
    {
            int x , y;
            for(x=1;x<=3;x++)
        {
            for(y=1; y<=3 ; y++)
            {
            System.out.println(x+"   "+y);
            }
        }
    }
}
```

Output:

```
1   1
1   2
1   3
2   1
2   2
2   3
3   1
3   2
3   3
```

Here x starts from 1 and then enters the inner loop where y too, starts from 1. That is why we get 1 and 1 as the first result. As told earlier the inner loop should finish its iteration first. Therefore y continues iteration till it reaches 3, in the meanwhile x continues to be at 1. That is why the next two outputs are 1 and 2 & 1 and 3. After it comes out of the inner loop the value of x increments by 1. Thus x is 2 and y starts back from 1 again and it reaches 3 while x continues to be 2. Hence the next result is 2 & 1, 2 & 2 and 2 & 3. Then x reaches 3 and y starts from 1, hence we get the result 3 & 1, 3 & 2 and 3 & 3. Since x is exhausted i.e. it has already iterated for 3 times, the control comes out of it and the program ends.

19. Name two jump statements and their use.

Ans. Jump statements in Java:

break- The break statement causes an immediate exit from the do-while, for, switch or while statement in which it appears.

continue- The continue statement is used with the loop instructions do-while, for, and while. It transfers control to the place in the program where the next iteration of the loop in which it appears, begins.

20. What is the difference between break and continue statements when they occur in a loop?

Ans. The break statement is used to exit from a loop. The continue statement is used skip the remaining statements in a loop and continue with the next iteration.

21. State one similarity and one difference between while and for loop.

Ans. **Similarity**: Both while and for loop is used as entry controlled loop.

Difference: The for loop may have initialisation, test expression and updation, written before the loop. The while loop have the initialisation always before the loop and updation in the body of the loop.

22. Analyze the following program segment and determine how many times the body of loop will be executed (show the working).

Ans. The loop executes as long as the given condition is true.

23. What is an exception?

Ans. An exception is a run-time error that occurs in a program.

24. Give the output and determine how many times the loop will execute:

```
x=1; y=1;
while(x<=y)
{
y = y/x;
System.out.println(y);
}
```

Ans. x=1; y=1;

x<=y -> the loop executes once.

y=y/x =1/1=1

x<=y -> the loop executes next time

The entire process will continue infinite number of times, with the output as 1 in different lines.

25. Give the output of the following code fragment:

```
when
(i) opn = 'b'
(ii) opn = 'x'
(iii) opn = 'a'
switch (opn)
{
case 'a':
      System.out.println("Platform Independent");
      break;
case 'b':
      System.out.println("Object Oreinted");
case 'c':
```

```
        System.out.println("Robust and Secure");
        default:
        System.out.println("Wrong Input");
    }
```

Ans. i. Object Oreinted

Robust and Secure
Wrong Input

ii. Wrong Input

iii. Platform Independent

26. Convert the following if else if construct into switch case

```
if( var==1)
System.out.println("good");
else if(var==2)
System.out.println("better");
else if(var==3)
System.out.println("best");
else
System.out.println("invalid");
```

Ans.

```
        switch(var)
{
case 1:
        System.out.println("good");
        break;
case 2:
        System.out.println("better");
        break;
case 3:
        System.out.println("best");
        break;
        default:
        System.out.println("invalid");
}
```

27. Rewrite the following using ternary operator:

```
if (bill >10000 )
discount = bill * 10.0/100;
else
discount = bill * 5.0/100;
```

Ans. discount=(bill >10000) ? bill * 10.0/100: bill * 5.0/100;

28. Rewrite the following program segment using the if .. else statement.

```
comm = (sale>15000)?sale*5/100:0;
```

Ans.

```
    if (sale>15000)
comm=sale*5/100
else
comm=0;
```

29. Rewrite the following using ternary operator :

```
if(x%2==0)
System.out.print("EVEN");
else
System.out.print("ODD");
```

Ans.

```
System.out.print((x%2==0)?"EVEN":"ODD");
```

30. Convert the following segment into an equivalent do loop.

```
int x,c;
for (x=10, c=20; c>=10; c = c – 2)
x++;
```

Ans.

```
int x=10,c=20;
do
{
    x++;
    c=c-2;
}while(c>=10);
```

31. Give the output of the following program segment and also mention how many times the loop is executed:

```
int i;
for (i = 5 : i > 10; i ++)
System.out.println(i);
System.out.println(i * 4);
```

Ans. Output:

20

The loop do not execute, as the condition results to false.

32. Convert the following while loop to the corresponding for loop:

```
int m = 5, n = 10;
while (n>=1)
{
System.out.println(m*n);
n–;
}
```

Ans.

```
int m=5,n;
for(n=10;n>=1;n--)
{
    System.out.println(m*n);
}
```

33. Convert following do-while loop into for loop.

```
int i = 1;
int d=5;
do {
d=d*2;
System.out.println(d;)
i++ ; } while (i<=5);
```

Ans.

```
int I,d=5;
for(i=1;i<=5;i++)
{
    d=d*2;
    System.out.println(d);
}
```

34. What will be the output of the following code?

```
int m=2;
int n=15;
for(int i = 1; i<5; i++);
m++; – –n;
System.out.println("m=" +m);
System.out.println("n=" +n);
```

Ans.

```
m=3
n=14
```

35. Analyze the given program segment and answer the following questions:

```
for(int i=3;i<=4;++)
{
for(int j=2;j<i;j++)
{
System.out.print(" ");
}
System.out.println("WIN");
}
```

i. How many times does the inner loop execute ?

ii. Write the output of the program segment.

Ans.

i. The inner loop executes 3 times.
ii. Output:
WIN
WIN

36. Analyse the following program segment and determine how many times the loop will be executed and what will be the output of the program segment.

```
int k=1, i=2;
while (++i<6)
k*=i;
System.out.println(k);
```

Ans. The loop executes 3 times.

Output:
60

37. How many times will the following loop execute? What value will be returned?

```
int x = 2, y = 50;
do
{
++x;
y- = x++;
}
while(x <= 10);
return y;
```

Ans. The loop will execute 5 times.

Value returned is 15.

38. Analyse the following program segment and determine how many times the loop will be executed and what will be the output of the program segment?

```
int p=200;
while(true)
{
if(p<100)
break;
p=p-20;
}
System.out.println(p);
```

Ans. The loop executes 7 times.

Output:
80

39. Name the types of error (syntax, runtime or logical error) in each case given below:

i. Division by a variable that contains a value of zero.
ii. Multiplication operator used when the operation should be division.

iii. Missing semicolon.

Ans.

i. Runtime error
ii. Logical error
iii. Syntax error

40. Give the output of the following program segment:

```
double x = 2.9, y = 2.5;
System.out.println(Math.min(Math.floor(x), y));
System.out.println(Math.max(Math.ceil(x), y));
```

Ans.

Output:
2.0
3.0

41. What is the final value of ctr after the iteration process given below, executes?

```
int ctr = 0;
for (int i = 1 ; i < = 5 ; i++)
for (int j = 1 ; j < = 5 ; j+=2)
++ctr;
```

Ans.

Final value of ctr is 15.

42. What are the final values stored in variables x and y below?

```
double a = - 6.35;
double b = 14.74;
double x = Math.abs(Math.ceil(a));
double y = Math.rint(Math.max(a,b));
```

Ans. x=6.0 and y=15.0

43. What are the values stored in variables r1 and r2:

(i) double r1 = Math.abs(Math.min(-2.83,-5.83));
(ii) double r2 = Math.sqrt(Math.floor(16.3));

Ans. r1=5.83 and r2=4.0

44. Rewrite the following program segment using if-else statements instead of the ternary operator.

String grade=(mark>=90) ? “A” : (mark>=80) ? “B” : “C”;

Ans.

```
String grade;
If(marks>=90)
    grade=“A”;
else
    grade=“B”;
```

45. Study the method and answer the given questions.

```
public void sampleMethod()
{
for(int i=0;i<3;i++)
{
for(int j=0;j<2;j++)
      {
          int number = (int) (Math.random( ) * 10);
          System.out.println(number);
      }
}
}
```

(i) How many times does the loop execute?
(ii) What is the range of possible values stored in the variable number?

Ans.

(i) The loops executes for 6 times.
(ii) The possible range of values stored in 'number' is 0 to 9.

46. What are the values of a and b after the following function is executed, if the values passed are 30 and 50:

```
void paws(int a, int b)
{
      a = a + b;
      b = a – b;
      a = a – b;
}
```

Ans.a=50 and b=30

47. Write the output of the following code segment:

```
char ch; int x = 97;
do
{
      ch = (char)x;
      System.out.print(ch+" ");
      if(x%10==0)
      break;
      ++x;
}
while(x<=100);
```

Ans. Output:

a b c d

48. Analyze the given program segment and answer the following questions:

(i) Write the output of the program segment.
(ii) How many times does the body of the loop gets executed?

```
for(int m=5; m<=20; m+=5)
{
    if(m%3 == 0)
    break;
    else
    if(m%5 == 0)
    System.out.println(m);
    continue;
}
```

Ans. i. Output:

5
10

ii. The loop executes 3 times.

49. Give the output of the following program segment and also mention the number of times the loop is executed:

```
int a,b;
for(a=6,b=4;a<=24;a=a+6)
{
          if(a%b==0)
        break;
}
System.out.println(a);
```

Ans. Output:

12

The loop executes 2 times.

50. Give the output of the following:

a) Math.floor (-4.7)
b) Math.ceil(3.4) + Math.pow(2, 3)
c) Math.floor(-126.349)
d) Math.max(45.6,17.3)
e) Math.min(-0.0,0.0)
f) Math.pow(4,3)
g) Math.sqrt(625)
h) Math.cbrt(125)
i) Math.max(11,11)
j) Math.ceil(-12.56)
k) Math.floor(15.36)
l) Math.round(146.5)
m) Math.max(-17, -19)
n) Math.ceil(7.8)
o) Math.ceil(4.2)
p) Math.abs(-4)

Ans. a) -5.0

b) 12.0
c) -127.0
d) 45.6
e) -0.0
f) 64.0
g) 25.0
h) 5.0
i) 11
j) -12.0
k) 15.0
l) 147
m) -17
n) 8.0
o) 5.0
p) 4

51. Write equivalent Java expressions for the following:

a. $\sqrt{a+b}$

b. $\frac{1}{3}a^3 + \frac{1}{3}b^3$

c. $s = ut + \frac{1}{2}at^2$

d. $d = \sqrt{l^2 + b^2}$

e. $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$

f. $\frac{a^3 + b^3}{a^3 - b^3}$

g. $\left|\frac{a-b}{a+b}\right|$

h. $\sqrt[3]{\frac{a+b}{ab}}$

i. $\frac{(a+b)^n}{\sqrt{3+b}}$

J. $z = \frac{5x^3 + 2y}{x+y}$

k. Under root of $2as+u^2$

l. $\frac{\sqrt{3x + x^2}}{a+b}$

m. $\frac{a^2 + b^2}{2ab}$

n. $\sqrt{T = A^2 + B^2} + C^2$

o. $ax^5 + bx^3 + c$

Ans.

a. Math.sqrt(a+b)
b. 1/3.0+Math.pow(a,3)+1/3.0*Math.pow(b,3)
c. s=u*t+1/2.0*a*Math.pow(t,2)
d. d=Math.sqrt(l*l+b*b);

e. (-b+Math.sqrt(b*b-4*a*c))/(2*a)
f. (Math.pow(a,3)+Math.pow(b,3))/ (Math.pow(a,3)-Math.pow(b,3))
g. Math.abs((a-b)/(a+b))
h. Math.cbrt((a+b)/(a*b))
i. Math.pow(a+b.n)/Math.sqrt(3+b)
j. z=(5*x*x*x+2*y)/(x+y);
k. Math.sqrt(2*a*s+u*u)
l. Math.sqrt(3*x+x*x)/(a+b)
m. (a*a+b*b)/(2*a*b)
n. T=Math.sqrt(A*A+B*B+C*C);
o. a*Math.pow(x,5)+b*Math.pow(x,3)+c

SECTION B

Programming Questions

1. Write a program to input the area of a square and find its perimeter.

Ans.
```
import java.util.*;
class Sol1
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      double a,s,p;
      System.out.println("Enter the area of a square:");
      a=sc.nextDouble();
      s=Math.sqrt(a);
      p=4*s;
      System.out.println("Perimeter="+p);
   }
}
```

2. Write a program to input the length and breadth of a rectangle and find its diagonal.

$$diagonal=\sqrt{length^2 + breadth^2}$$

Ans.
```
import java.util.*;
class Sol2
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      double l,b,d;
```

```
        System.out.println("Enter the length and breadth of the rectangle:");
        l=sc.nextDouble();
        b=sc.nextDouble();
        d=Math.sqrt(l*l+b*b);
        System.out.println("Diagonal="+d);
    }
}
```

3. Write a program to input 2 integers and check whether both the numbers are multiples of 7 or not.

Ans.

```
import java.util.*;
class Sol3
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a,b;
        System.out.println("Enter 2 integers:");
        a=sc.nextInt();
        b=sc.nextInt();
        if(a%7==0 && b%7==0)
            System.out.println("Both are multiples of 7");
        else
            System.out.println("Both are not multiples of 7");
    }
}
```

4. Write a program to pass 2 integer numbers as parameters. If either of the two numbers is 0, display invalid entry and the program should end, if it is valid entry, divide the larger number with the smaller number and display the result.

Ans.

```
class Sol4
{
    static void divide(int a,int b)
    {
      if(a==0 || b==0)
       System.out.println("Invalid Entry");
      else
      {
         float q;
         if(a>b)
          q=(float)a/b;
         else
          q=(float)b/a;
         System.out.println("Quotient:"+q);
```

```
    }
  }
}
```

5. Write a program to input 3 unique integers and print the smallest among them.

Ans.

```
import java.util.*;
class Sol5
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a,b,c;
    System.out.println("Enter 3 integers:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    if(a<b && a<c)
      System.out.println("Smallest:"+a);
    else if(b<a && b<c)
      System.out.println("Smallest:"+b);
    else
      System.out.println("Smallest:"+c);
  }
}
```

6. Write a program to input the three angles of a triangle and check whether it forms a triangle or not, if it forms a triangle, check whether it is an equilateral, isosceles or a scalene triangle.

(Hint: To form a triangle, the sum of the angles should be 180 degrees.
To form an equilateral triangle every angle should be equal.
To form an isosceles triangle any two angles should be equal.
To form a scalene triangle all three angles should be different from each other.)

Ans.

```
import java.util.*;
class Sol6
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a,b,c;
    System.out.println("Enter 3 angles:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    if(a+b+c==180)
```

```
    {
      if(a<90 && b<90 && c<90)
        System.out.println("Acute angled triangle");
      else if(a>90 || b>90 || c>90)
        System.out.println("Obtuse angled triangle");
      else
        System.out.println("Right angled triangle");
    }
    else
        System.out.println("Cannot form a triangle");
  }
}
```

7. Write a program to input the three sides of a triangle and check whether it forms a triangle or not, if it forms a triangle, check whether it is an equilateral, isosceles or a scalene triangle.

(Hint: To form a triangle, each side should be less the sum of the other two sides.
To form an equilateral triangle every side should be equal.
To form an isosceles triangle any two sides should be equal.
To form a scalene triangle all three sides should be different from each other.)

Ans.

```
import java.util.*;
class Sol7
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a,b,c;
    System.out.println("Enter 3 sides:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    if(a<b+c && b<a+c && c<a+b)
    {
      if(a==b && b==c)
        System.out.println("Equilateral triangle");
      else if(a==b || b==c || c==a)
        System.out.println("Isosceles triangle");
      else
        System.out.println("Scalene triangle");
    }
    else
        System.out.println("Cannot form a triangle");
  }
}
```

8. Write a program to accept three sides of a triangle as parameter and check whether it can form a triangle or not. If it forms a triangle, check whether it is an acute angled, obtuse angled or right-angled triangle.

(Hint: To form a triangle, each side should be less the sum of the other two sides..

To form an acute angled triangle the square of every side should be less than the sum of the squares of the other two sides.

To form an obtuse angled triangle the square of any side should be greater than the sum of the squares of the other two sides.

To form an right angled triangle the square of any side should be equal to the sum of the

squares of the other two sides.)

Ans.

```
import java.util.*;
class Sol8
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a,b,c;
      System.out.println("Enter 3 sides:");
      a=sc.nextInt();
      b=sc.nextInt();
      c=sc.nextInt();
      if(a<b+c && b<a+c && c<a+b)
      {
         if(a*a<b*b+c*c && b*b<a*a+c*c && c*c<a*a+b*b)
            System.out.println("Acute angled triangle");
         else if(a*a>b*b+c*c || b*b>a*a+c*c || c*c>a*a+b*b)
            System.out.println("Obtuse angled triangle");
         else
            System.out.println("Right angled triangle");
      }
      else
            System.out.println("Cannot form a triangle");
   }
}
```

9. Write a program to accept a mark obtained by a student in computer science and print the grades accordingly:

Marks	Grade
Above 90	A
70 to 89	B
50 to 69	C
below 50	D

Ans.
```
import java.util.*;
class Sol9
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int c;
      System.out.println("Enter marks in Computer science:");
      c=sc.nextInt();
      if(c>=90)
        System.out.println("Grade=A");
      else if(c>=70 && c<90)
        System.out.println("Grade=B");
      else if(c>=50 && c<70)
        System.out.println("Grade=C");
      else
        System.out.println("Grade=D");
   }
}
```

10. A cloth showroom has announced the following festival discounts on the purchase of items, based on the total cost of the items purchased:

Total Cost	Discount (in Percentage)
Less than ₹2000	5%
₹2001 to ₹5000	25%
₹5001 to ₹10000	35%
Above ₹10000	50%

Write a program to input the total cost and compute and display the amount to be paid by the customer after availing the discount.

Ans.
```
import java.util.*;
class Sol10
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      float tc,d,ap;
      System.out.println("Enter the total cost of the items:");
      tc=sc.nextFloat();
      if(tc<=2000)
        d=5/100f*tc;
      else if(tc>=2001 && tc<=5000)
        d=25/100f*tc;
      else if(tc>=5001 && tc<=10000)
```

```
        d=35/100f*tc;
      else
        d=50/100f*tc;
      ap=tc-d;
      System.out.println("Amount Payable:"+ap);
    }
}
```

11. An electronics shop has announced the following seasonal discounts on the purchase of certain items.

Purchase Amount is₹	Discount on Laptop	Discount on desktop PC
0–2500	0.0%	5.0%
25001 – 57000	5.0%	7.5%
57001 – 100000	7.5%	10.0%
More than 100000	10.0%	15.0%

Write a program based on the above criteria, to input name, address, amount of purchase and the type of purchase (L for Laptop and D for Desktop) by a customer. Compute and print the net amount to be paid by a customer along with his name and address.

(Hint: discount = (discount rate/100)* amount of purchase

Net amount = amount of purchase – discount)

Ans.

```
import java.util.*;
class Sol11
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String name, add;
      float a,d=0,na=0;
      char type;
      System.out.println("Enter the name:");
      name=sc.nextLine();
      System.out.println("Enter the address:");
      add=sc.nextLine();
      System.out.println("Enter the amount of purchase:");
      a=sc.nextFloat();
      System.out.println("Enter the type of purchase:");
      type=sc.next().charAt(0);
      if(type=='L')
      {
        if(a>=0 && a<=25000)
          d=0;
```

```
        else if(a>=25001 && a<=57000)
            d=5/100f*a;
        else if(a>=57001 && a<=100000)
            d=7.5f/100*a;
        else
            d=10/100f*a;
      }
      if(type==’D’)
      {
        if(a>=0 && a<=25000)
            d=5/100f*a;
        else if(a>=25001 && a<=57000)
            d=7.5f/100*a;
        else if(a>=57001 && a<=100000)
            d=10f/100*a;
        else
            d=15/100f*a;
      }
      na=a-d;
      System.out.println(“Name:”+name);
      System.out.println(“Address:”+add);
      System.out.println(“Ndet amount:”+na);
    }
}
```

12. Given below is a hypothetical table showing rates of Income Tax for male citizens below the age of 65 years:

Taxable Income (TI) in	Income Tax in
Does not exceed 1,60,000	Nil
Is greater than 1,60,000 and less than or equal to 5,00,000	(TI – 1,60,000) * 10%
Is greater than 5,00,000 and less than or equal to 8,00,000	[(TI – 5,00,000) * 20%] + 34,000
Is greater than 8,00,000	[(TI – 8,00,000) * 30%] + 94,000

Write a program to input the age, gender (male or female) and Taxable Income of a person. If the age is more than 65 years or the gender is female, display “wrong category*.

If the age is less than or equal to 65 years and the gender is male, compute and display the Income Tax payable as per the table given above.

Ans.

```
import java.util.*;
class Sol12
{
   static void main()
```

```
    {
        Scanner sc=new Scanner(System.in);
        int age;
        String gender;
        float ti,it;
        System.out.println("Enter the age:");
        age=sc.nextInt();
        System.out.println("Enter the gender:");
        gender=sc.nextLine();
        sc.nextLine();//dummy input
        System.out.println("Enter the taxable income:");
        ti=sc.nextFloat();
        if(age>65 && gender.equals("female"))
            System.out.println("Wrong category");
        else
        {
            if(ti<=160000)
                it=0;
            else if(ti>160000 && ti<=500000)
                it=10/100f*(ti-160000);
            else if(ti>500000 && ti<=800000)
                it=20/100f*(ti-500000)+34000;
            else
                it=30/100f*(ti-800000)+94000;
            System.out.println("Income Taxable:"+it);
        }
    }
}
```

13. Write programs for each of the following to print the series:

a. 2, 4, 6, 8, 10, … , 100
b. 99, 97, 95, 93, 91, …, 1
c. 7, 14, 21, 28, 35, …, 70
d. 80, 72, 64, 56, 48, …, 8
e. 1, 4, 9, 16, 25, 36, …. 100
f. 0, 3, 8, 15, 24, 35, …, 99
g. 1, 2, 4, 7, 11, 16, 22, 29, …, upto n terms. [Take n as input]
h. 2, 4, 8, 14, 22, 32, 44, 59, …, upto n terms. [Take n as input]
i. 1, 2, 5, 10, 17, 26, 37, 50, …, upto n terms. [Take n as input]
j. 1, 1, 2, 3, 5, 8, 13, 21, 34, …, upto n terms. [Take n as input]
k. 1, 2, 5, 12, 29, 70, 169, …, upto n terms. [Take n as input]

Ans.

a)

```
class Sol13
{
```

```
    static void main()
    {
        int i;
        for(i=2;i<=100;i+=2)
        {
            System.out.print(i+" ");
        }
    }
}

b)
class Sol13
{
    static void main()
    {
        int i;
        for(i=99;i>=1;i-=2)
        {
            System.out.print(i+" ");
        }
    }
}

c)
class Sol13
{
    static void main()
    {
        int i;
        for(i=7;i<=70;i+=7)
        {
            System.out.print(i+" ");
        }
    }
}

d)
class Sol13
{
    static void main()
    {
        int i;
        for(i=80;i>=8;i-=8)
        {
```

```
            System.out.print(i+" ");
        }
    }
}

e)
class Sol13
{
    static void main()
    {
        int i;
        for(i=1;i<=10;i++)
        {
            System.out.print((i*i)+" ");
        }
    }
}

f)
class Sol13
{
    static void main()
    {
        int i;
        for(i=1;i<=10;i++)
        {
            System.out.print((i*i-1)+" ");
        }
    }
}

g)
import java.util.*;
class Sol13
{
    static void main()
    {
        int i,n,s=1;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of terms:");
        n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            System.out.print(s+" ");
```

```
            s=s+i;
        }
    }
}
```

h)

```
import java.util.*;
class Sol13
{
    static void h()
    {
        int i,n,s=2,c=2;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of terms:");
        n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            System.out.print(s+" ");
            s=s+c;
            c=c+2;
        }
    }
}
```

i)

```
import java.util.*;
class Sol13
{
    static void main()
    {
        int i,n,s=1,c=1;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of terms:");
        n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            System.out.print(s+" ");
            s=s+c;
            c=c+2;
        }
    }
}
```

j)

```
import java.util.*;
class Sol13
{
   static void main()
   {
      int i,n,f=1,s=0,t;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the number of terms:");
      n=sc.nextInt();
      for(i=1;i<=n;i++)
      {
         t=f+s;
         System.out.print(t+" ");
         f=s;
         s=t;
      }
   }
}
```

k)

```
import java.util.*;
class Sol13
{
   static void main()
   {
      int i,n,f=1,s=0,t;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the number of terms:");
      n=sc.nextInt();
      for(i=1;i<=n;i++)
      {
         t=f+2*s;
         System.out.print(t+" ");
         f=s;
         s=t;
      }
   }
}
```

14. Write a program to find the sum of all 3-digit even natural numbers.

Ans.

```
class Sol14
{
   static void main()
```

```
    {
      int i,s=0;
      for(i=100;i<=998;i+=2)
      {
        s+=i;
      }
      System.out.print("Sum="+s);
    }
}
```

15. Write a program to find the sum of all 3 digit odd natural numbers, which are multiples of 5.

Ans.

```
class Sol15
{
   static void main()
   {
      int i,s=0;
      for(i=101;i<=999;i+=2)
      {
         If(i%5==0)
s+=i;
      }
      System.out.print("Sum="+s);
   }
}
```

16. Write a program to input an integer and find its factorial. Factorial of a number is the product of all natural numbers till that number. For example factorial of 5 is 120 since 1×2×3×4×5=120.

Ans.

```
import java.util.*;
class Sol16
{
   static void main()
   {
      long i,n,f=1;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter an integer:");
      n=sc.nextInt();
      for(i=1;i<=n;i++)
      {
         f=f*i;
      }
      System.out.print("Factotrial:"+f);
   }
}
```

17. Write a program to input an integer and check whether it is a prime number or not.

Ans.

```
import java.util.*;
class Sol17
{
   static void main()
   {
      long i,n,c=0;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter an integer:");
      n=sc.nextInt();
      for(i=1;i<=n;i++)
      {
         If(n%i==0)
      c++;
      }
      If(c==2)
           System.out.print("Prime Number");
      else
      System.out.print("Not a Prime Number");
   }
}
```

18. Write a program to input 10 integers and find the sum of two-digit as well as three-digit numbers separately.

Ans.

```
import java.util.*;
class Sol18
{
   static void main()
   {
      int i,n,s2=0,s3=0;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter 10 integers:");
      for(i=1;i<=10;i++)
      {
           n=sc.nextInt();
      if(n>=10 && n<=99)
           s2+=n;
      if(n>=100 && n<=999)
           s3+=n;
      }
      System.out.println("Sum of 2 digit numbers:"+s2);
      System.out.println("Sum of 3 digit numbers:"+s3);
   }
}
```

19. Write a program to input 10 integers and display the largest as well as the smallest integer.

Ans.

```
import java.util.*;
class Sol19
{
   static void main()
   {
      int i,n,lar=0,sma=0;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter 10 integers:");
      for(i=1;i<=10;i++)
      {
           n=sc.nextInt();
      if(i==1)
           lar=sma=n;
      else
      {
           if(n>lar)
                lar=n;
           if(n<sma)
                sma=n;
      }
      }
      System.out.println("Largest number:"+lar);
      System.out.println("Smallest number:"+sma);
   }
}
```

20. Write a program to input 10 integers and check whether all the entered numbers are even numbers or not.

Ans.

```
import java.util.*;
class Sol20
{
   static void main()
   {
      int i,n,f=0;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter 10 integers:");
      for(i=1;i<=10;i++)
      {
           n=sc.nextInt();
      if(n%2!=0)
           f=1;
      }
      if(f==0)
```

```
            System.out.println("All are even numbers");
        else
            System.out.println("All are not even numbers");
    }
}
```

21. Write a program to input 10 integers and check whether all the entered numbers are same or not.

For Example,
INPUT:
Enter 10 numbers: 10 12 13 234 45 34 67 78 76 12
OUTPUT:
All numbers are not same.
INPUT:
Enter 10 numbers: 12 12 12 12 12 12 12 12 12 12
OUTPUT:
All numbers are same.
Ans.

```
import java.util.*;
class Sol20
{
   static void main()
   {
      int i,n,f=0,p=0;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter 10 integers:");
      for(i=1;i<=10;i++)
      {
          n=sc.nextInt();
      if(i==1)
          p=n;
      else
      {
          if(n!=p)
                f=1;
      }
      }
      if(f==0)
          System.out.println("All numbers are same");
      else
          System.out.println("All numbers are not same");
   }
}
```

22. Write a program to input 10 integers and check whether the entered numbers are in ascending order or not.

For Example,
INPUT:
Enter 10 numbers: 10 12 13 25 45 55 67 78 106 122
OUTPUT:
The numbers are in ascending order.
INPUT:
Enter 10 numbers: 25 34 56 67 12 32 43 21 23 111
OUTPUT:
The numbers are not in ascending order.

Ans.

```
import java.util.*;
class Sol22
{
   static void main()
   {
      int i,n,f=0,p=0;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter 10 integers:");
      for(i=1;i<=10;i++)
      {
         n=sc.nextInt();
         if(i==1)
            p=n;
         else
         {
            if(n<p)
               f=1;
            p=n;
         }
      }
      if(f==0)
         System.out.println("The numbers are in ascending order");
      else
           System.out.println("The numbers are not in ascending order");
   }
}
```

23. Write a program to print the first 15 numbers of the Pell series. Pell series is such a series which starts from 1 and 2 , and subsequent numbers is the sum of twice the previous number and the number previous to the previous number. Pell series: 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, 13860, …

Ans.

```
class Sol23
{
```

```
    static void main()
    {
      int i,n,f=1,s=0,t;
      for(i=1;i<=15;i++)
      {
        t=f+2*s;
        System.out.print(t+" ");
        f=s;
        s=t;
      }
    }
}
```

24. Write a program to find the sum of 1st 10 numbers of Lucas series i.e. 2,1,3,4,7,11,18,…. Lucas series is such a series which starting from 2 and 1, and subsequent numbers are the sum of the previous two numbers.

Ans.

```
class Sol24
{
  static void main()
  {
    int i,n,f=2,s=1,t,sum=0;
    sum=f+s;
    for(i=1;i<=8;i++)
    {
      t=f+s;
      sum+=t;
      f=s;
      s=t;
    }
    System.out.println("Sum="+sum);
  }
}
```

25. Write a program to input an integer and check whether it is perfect, abundant or deficient number. If the sum of the factors excluding itself is equal to that number it is perfect, if greater than that number it is abundant and if less than that number it is deficient number.

Ans.

```
import java.util.*;
class Sol25
{
  static void main()
  {
    int i,n,s=0;
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter an integer:");
```

```
        n=sc.nextInt();
        for(i=1;i<n;i++)
        {
           if(n%i==0)
             s+=i;
        }
        if(s==n)
           System.out.println("Perfect Number");
        else if(s>n)
           System.out.println("Abundant Number");
        else
            System.out.println("Deficient Number");
    }
}
```

26. Write a program to input two integers and check whether it forms an amicable pair or not. An amicable pair is such that, the sum of the factors excluding itself of one number is the other number and sum of the factors excluding itself of the other number is this number.

Example, (220, 284). Since sum of factors excluding itself of :
220= 1+2+4+5+10+11+20+22+ 44+55+110=284
284= 1+ 2+4+71+142=220.

Ans.

```
import java.util.*;
class Sol26
{
   static void main()
   {
      int a,b,sa=0,sb=0;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter 2 integers:");
      a=sc.nextInt();
      b=sc.nextInt();
      for(i=1;i<a;i++)
      {
         if(a%i==0)
           sa+=i;
      }
      for(i=1;i<b;i++)
      {
         if(b%i==0)
           sb+=i;
      }

      if(sa==b && sb==a)
         System.out.println("Amicable Pair");
```

```
        else
            System.out.println("Not an Amicable Pair");
    }
}
```

27. Using nested loops write programs to generate the following patterns on the screen:

i.
```
1
12
123
1234
12345
```

ii.
```
5
54
543
5432
54321
```

iii.
```
1
21
321
4321
54321
```

iv.
```
54321
5432
543
54
5
```

v.
```
54321
4321
321
21
1
```

vi.
```
1
22
333
4444
55555
```

vii.
```
55555
4444
333
22
1
```

viii.
```
11111
2222
333
44
5
```

ix.
```
5
44
333
2222
11111
```

x.
```
*
**
***
****
*****
```

xi.
```
*****
****
***
**
*
```

xii
```
*           *
    *   *
      *
    *   *
*           *
```

xiii.
```
1
01
101
0101
10101
```

xiv.
```
12345
2345
345
45
5
```

xv
```
1
01
010
1010
10101
010101
```

xvi.
```
1
01
101
1001
10001
100001
```

xvii.
```
1
11
112
1123
11235
112358
```

xviii.
```
1
121
12321
1234321
123454321
12345654321
```

xix.
```
65432123456
 543212345
  4321321
   32123
    212
     1
```

xx.
```
101010
010101
101010
010101
101010
```

xxi.
```
12345
23451
34512
45123
51234
```

xxii.
```
54321
43215
32154
21543
15432
```

xxiii.
```
12345
51234
45123
34512
23451
```

xxiv.

xxv.

xxvi.
```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 5 1
1 6 15 15 6 1
```

xxvii.
```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 14
```

Ans.

i)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=1;j<=i;j++)
         {
            System.out.print(j);
         }
         System.out.println();
      }
   }
}
```

ii)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=5;i>=1;i--)
      {
         for(j=5;j>=i;j--)
         {
            System.out.print(j);
         }
         System.out.println();
      }
   }
}
```

iii)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=i;j>=1;j--)
         {
            System.out.print(j);
         }
         System.out.println();
      }
   }
}
```

iv)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=5;j>=i;j--)
```

```
            {
               System.out.print(j);
            }
            System.out.println();
         }
      }
   }
```

v)

```
   class Sol27
   {
      static void main()
      {
         int i,j;
         for(i=5;i>=1;i--)
         {
            for(j=i;j>=1;j--)
            {
               System.out.print(j);
            }
            System.out.println();
         }
      }
   }
```

vi)

```
   class Sol27
   {
      static void main()
      {
         int i,j;
         for(i=1;i<=5;i++)
         {
            for(j=1;j<=i;j++)
            {
               System.out.print(i);
            }
            System.out.println();
         }
      }
   }
```

vii)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=5;i>=1;i--)
      {
         for(j=1;j<=i;j++)
         {
            System.out.print(i);
         }
         System.out.println();
      }
   }
}
```

viii)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=i;j<=5;j++)
         {
            System.out.print(i);
         }
         System.out.println();
      }
   }
}
```

ix)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=5;i>=1;i--)
      {
         for(j=i;j<=5;j++)
         {
```

```
            System.out.print(i);
        }
        System.out.println();
    }
  }
}
```

x)

```
class Sol27
{
  static void main()
  {
    int i,j;
    for(i=1;i<=5;i++)
    {
      for(j=1;j<=i;j++)
      {
        System.out.print("*");
      }
      System.out.println();
    }
  }
}
```

xi)

```
class Sol27
{
  static void main()
  {
    int i,j;
    for(i=5;i>=1;i--)
    {
      for(j=1;j<=i;j++)
      {
        System.out.print("*");
      }
      System.out.println();
    }
  }
}
```

xii)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=1;j<=5;j++)
         {
            if(i==j || i+j==6)
System.out.print("*");
      else
System.out.print(" ");
         }
         System.out.println();
      }
   }
}
```

xiii)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=1;j<=i;j++)
         {
            if((i+j)%2==0)
System.out.print("1");
      else
System.out.print("0");
         }
         System.out.println();
      }
   }
}
```

xiv)

```
class Sol27
{
```

```
    static void main()
    {
       int i,j;
       for(i=1;i<=5;i++)
       {
          for(j=i;j<=5;j++)
          {
System.out.print(j);
          }
          System.out.println();
       }
    }
}
```

xv)

```
class Sol27
{
   static void main()
   {
      int i,j,c=1;
      for(i=1;i<=6;i++)
      {
         for(j=1;j<=i;j++)
         {
            if(c%2==0)
System.out.print("0");
      else
System.out.print("1");
      c++;
         }
         System.out.println();
      }
   }
}
```

xvi)

```
class Sol27
{
   static void main()
   {
      int i,j;
      for(i=1;i<=6;i++)
      {
         for(j=1;j<=i;j++)
```

```
            {
               if(i==j || j==1)
        System.out.print("*");
            else
        System.out.print(" ");
            }
            System.out.println();
          }
        }
      }
```

xvii)

```
      class Sol27
      {
        static void main()
        {
          int i,j,f,s,t;
          for(i=1;i<=6;i++)
          {
            f=1;s=0;
            for(j=1;j<=i;j++)
            {
              t=f+s;
              System.out.print(t);
              f=s;s=t;
            }
            System.out.println();
          }
        }
      }
```

xviii)

```
      class Sol27
      {
        static void main()
        {
          int i,j,f;
          for(i=1;i<=6;i++)
          {
            f=0;
            for(j=1;j<=2*i-1;j++)
            {
              if(j<=i)
                ++f;
```

```
                else
                   --f;
                System.out.print(f);
             }
             System.out.println();
          }
       }
    }
```

xix)

```
    class Sol27
    {
       static void main()
       {
          int i,j;
          for(i=6;i>=1;i--)
          {
             for(j=i;j<=5;j++)
                System.out.print(" ");
             for(j=i;j>=1;j--)
                System.out.print(j);
             for(j=2;j<=i;j++)
                System.out.print(j);
             System.out.println();
          }
       }
    }
```

xx)

```
    class Sol27
    {
       static void main()
       {
          int i,j;
          for(i=1;i<=5;i++)
          {
               for(j=1;j<=5;j++)
    {
          if((i+j)%2==0)
          System.out.print("1");
          else
          System.out.print("0");
    }
             System.out.println();
```

```
        }
      }
    }
```

xxi)

```
    class Sol27
    {
      static void main()
      {
        int i,j;
        for(i=1;i<=5;i++)
        {
          for(j=i;j<=5;j++)
            System.out.print(j);
          for(j=1;j<=i-1;j++)
            System.out.print(j);
          System.out.println();
        }
      }
    }
```

xxii)

```
    class Sol27
    {
      static void main()
      {
        int i,j;
        for(i=5;i>=1;i--)
        {
          for(j=i;j>=1;j--)
            System.out.print(j);
          for(j=5;j>=i+1;j--)
            System.out.print(j);
          System.out.println();
        }
      }
    }
```

xxiii)

```
    class Sol27
    {
      static void main()
      {
        int i,j;
```

```
            for(i=5;i>=1;i--)
            {
               for(j=i+1;j<=5;j++)
                  System.out.print(j);
               for(j=1;j<=i;j++)
                  System.out.print(j);
               System.out.println();
            }
         }
      }
```

xxiv)

```
      class Sol27
      {
         static void main()
         {
            int i,j;
            for(i=1;i<=6;i++)
            {
               for(j=1;j<=11;j++)
               {
                  if(i+j>=7 && j-i<=5)
                  {
                     if((i+j)%2==0)
                        System.out.print(" ");
                     else
                        System.out.print("*");
                  }
                  else
                        System.out.print(" ");
               }
               System.out.println();
            }
         }
      }
```

xxv)

```
      class Sol27
      {
         static void main()
         {
            int i,j,f;
            for(i=1;i<=6;i++)
            {
```

```
            for(j=1;j<=11;j++)
            {
               if(i+j>=7 && j-i<=5)
                  System.out.print("*");
               else
                  System.out.print(" ");
            }
            System.out.println();
         }
      }
   }
```

xxvi)

```
   class Sol27
   {
      static void main()
      {
         int i,j,f;
         for(i=0;i<7;i++)
         {
            f=1;
            for(j=0;j<=i;j++)
            {
               System.out.print(f+" ");
               f=f*(i-j)/(j+1);
            }
            System.out.println();
         }
      }
   }
```

xxvii)

```
   class Sol27
   {
       static void main()
      {
         int i,j,f=1;
         for(i=1;i<=5;i++)
         {
    for(j=1;j<=i;j++)
             {
                    System.out.print(f+" ");
                    f++;
            }
```

```
            System.out.println();
        }
    }
}
```

28. Write a program to input 10 numbers and print the largest prime number if any.

Ans.

```
import java.util.*;
class Sol28
{
    static void main()
    {
        int i,n,f=0,c,j,max=0;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter 10 integers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            c=0;
            for(j=1;j<=n;j++)
            {
                if(n%j==0)
                    c++;
            }
            if(c==2)
            {
                if(f==0)
                {
                    max=n;
                    f=1;
                }
                else
                {
                    if(n>max)
                        max=n;
                }
            }
        }
        if(f==0)
            System.out.println("No prime numbers found");
        else
            System.out.println("Largest prime number:"+max);
    }
}
```

29. Write a program to generate all 2 digit twisted prime number. Twisted prime number is a number which is prime and its reverse is also prime. Example 13 and 31.

Ans.
```
class Sol29
{
   static void main()
   {
      int i,j,r,c1,c2,d;
      for(i=10;i<=99;i++)
      {
         c1=0;
         for(j=1;j<=i;j++)
         {
            if(i%j==0)
               c1++;
         }
         r=0;
         for(j=i;j>0;j/=10)
         {
            d=j%10;
            r=r*10+d;
         }
         c2=0;
         for(j=1;j<=r;j++)
         {
            if(r%j==0)
               c2++;
         }
         if(c1==2  && c2==2)
            System.out.println(i+" ");
      }
   }
}
```

30. Input an integer and form a new number where the digits are in ascending order.

Ans.
```
import java.util.*;
class Sol30
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,j,n,s=0,c=0,d;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=9;i>=0;i--)
```

```
    {
      for(j=n;j>0;j/=10)
      {
        d=j%10;
        if(d==i)
          s=s+d*(int)Math.pow(10,c++);
      }
    }
    System.out.println("New Number:"+s);
  }
}
```

31. Write a program to pass an integer as argument and print the largest as well as smallest digit.

Ans.

```
class Sol31
{
  static void main(int n)
  {
    int i,d,max=0,min=0;
    for(i=n;i>0;i/=10)
    {
      d=i%10;
      if(max==0)
        max=d;
      if(min==0)
        min=d;
      if(d>max)
        max=d;
      if(d<min)
        min=d;
    }
    System.out.println("Largest digit:"+max);
    System.out.println("Smallest digit:"+min);
  }
}
```

32. Write a program to pass an integer as argument and print the second largest digit.

Ans.

```
class Sol32
{
  static void main(int n)
  {
    int i,d,lar=0,slar=0;
    for(i=n;i>0;i/=10)
    {
```

```
            d=i%10;
            if(lar==0)
               lar=d;
            if(d>lar)
               lar=d;
         }
         for(i=n;i>0;i/=10)
         {
            d=i%10;
            if(d!=lar)
            {
               if(slar==0)
                  slar=d;
               if(d>slar)
                  slar=d;
            }
         }
         System.out.println("Second Largest digit:"+slar);
      }
   }
```

33. Write a program to input an integer and remove all the even digits from it.

For Example,
INPUT: Enter an integer: 1234
OUTPUT:
13

Ans.

```
import java.util.*;
class Sol33
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,d,n,s=0,c=0;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=n;i>0;i/=10)
      {
         d=i%10;
         if(d%2==1)
            s=s+d*(int)Math.pow(10,c++);
      }
      System.out.println("New Number:"+s);
   }
```

```
}
```

34. Write a program to pass an integer as argument and print the number by having the digits arranged in ascending order.

Ans.

```
class Sol34
{
   static void main(int n)
   {
      int i,j,s=0,c=0,d;
      for(i=9;i>=0;i--)
      {
         for(j=n;j>0;j/=10)
         {
            d=j%10;
            if(d==i)
               s=s+d*(int)Math.pow(10,c++);
         }
      }
      System.out.println("New Number:"+s);
   }
}
```

35. Write a program to pass an integer as argument and check whether all digits in it are unique or not.

Ans.

```
class Sol35
{
   static void main(int n)
   {
      int i,j,f=0,d;
      for(i=n;i>0;i/=10)
      {
         for(j=i/10;j>0;j/=10)
         {
            if(i%10==j%10)
               f=1;
         }
      }
      if(f==0)
         System.out.println("All digits are unique");
      else
         System.out.println("All digits are not unique");
   }
}
```

36. Write a program to pass an integer as argument and print the frequency of each digit in it.

Ans.

```
class Sol36
{
   static void main(int n)
   {
      int i,j,f=0;
      for(i=0;i<=9;i++)
      {
         f=0;
         for(j=n;j>0;j/=10)
         {
            if(i==j%10)
               f++;
         }
         if(f>0)
         System.out.println("Frequency of "+i+" is="+f);
      }
   }
}
```

37. Write a program to input an integer and check whether it is an automorphic, trimorphic or triautomorphic number or not. A number n is said to be automorphic, if its square ends in n. For instance 5 is automorphic, because 52= 25, which ends in 5, 25 is automorphic, because 252=625, which ends in 25. A number n is called trimorphic if n3 ends in n. For example 493, = 117649, is trimorphic. A number n is called tri-automorphic if 3n2 ends in n; for example 667 is tri-automorphic because 3 × 6672, = 1334667, ends in 667.

Ans.

```
import java.util.*;
class Sol37
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,j,f=0,n;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=n;i>0;i/=10)
         f++;
      if((n*n)%(int)Math.pow(10,f)==n)
         System.out.println("Automorphic Number");
      else if((n*n*n)%(int)Math.pow(10,f)==n)
         System.out.println("Trimorphic Number");
      else if((3*n*n)%(int)Math.pow(10,f)==n)
         System.out.println("Tri-automorphic Number");
       else
```

```
            System.out.println("Not Automorhic,Trimorphic or Tri-automorphic Number");
        }
    }
```

38. Write a program to input a number and check whether it is a happy number or not. If you iterate the process of summing the squares of the decimal digits of a number and if this process terminates in 1, then the original number is called a happy number. For example 7=> (72)=49=> (42+92)=97=>(92 +72)=130 =>(12 +32+02)=10 =>(12+02)= 1.

Ans.

```
import java.util.*;
class Sol38
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int d,s,n;
        System.out.println("Enter a number:");
        n=sc.nextInt();
        do
        {
            s=0;
            while(n>0)
            {
                d=n%10;
                s+=d*d;
                n/=10;
            }
            n=s;
        }while(n>9);
        if(n==1)
            System.out.println("Happy Number");
        else
            System.out.println("Not a Happy Number");
    }
}
```

39. Write a program to input a number and check whether it is a magic number or not. If you iterate the process of summing the squares of the decimal digits of a number and if this process terminates in 1, then the original number is called a magic number. For example 55=> (5+5)=10=>(1+0)=1.

Ans.

```
import java.util.*;
class Sol39
{
    static void main()
    {
```

```
        Scanner sc=new Scanner(System.in);
        int d,s,n;
        System.out.println("Enter a number:");
        n=sc.nextInt();
        do
        {
          s=0;
          while(n>0)
          {
            d=n%10;
            s+=d;
            n/=10;
          }
          n=s;
        }while(n>9);
        if(n==1)
          System.out.println("Magic Number");
        else
          System.out.println("Not a Magic Number");
      }
    }
```

40. Write a program to input an integer and check whether it is Harshad or Niven number or not. A number is said to be Harshad if it is divisible by the sum of the digits of that number, example 126 and 1729.

Ans.

```
import java.util.*;
class Sol40
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int d,s=0,n,i;
    System.out.println("Enter a number:");
    n=sc.nextInt();
    for(i=n;i>0;i/=10)
    {
      d=i%10;
      s+=d;
    }
    if(n%s==0)
      System.out.println("Harshad Number");
    else
      System.out.println("Not a Harshad Number");
  }
}
```

41. Write a program to input a number and check whether it is a Kaprekar number or not. Take a positive whole number n that has d number of digits. Take the square n and separate the result into two pieces: a right-hand piece that has d digits and a left-hand piece that has either d or d-1 digits. Add these two pieces together. If the result is n, then n is a Kaprekar number. Examples are 9 (92 = 81, 8 + 1 = 9), 45 (452 = 2025, 20 + 25 = 45), and 297 (2972 = 88209, 88 + 209 = 297).

Ans.

```
import java.util.*;
class Sol41
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int f=0,n,i,h1,h2;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=n;i>0;i/=10)
         f++;
      h1=n/(int)Math.pow(10,f);
      h2=n%(int)Math.pow(10,f);
      if(h1+h2==n)
         System.out.println("Kaprekar Number");
      else
         System.out.println("Not a Kaprekar Number");
   }
}
```

42. Write a program to input two integers and find their Least Common Multiple(L.C.M).

For Example,
INPUT:
Enter 2 integers:
12
8
OUTPUT:
L.C.M. = 24

Ans.

```
import java.util.*;
class Sol42
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,a,b,l=0;
      System.out.println("Enter 2 numbers:");
      a=sc.nextInt();
      b=sc.nextInt();
```

```
        for(i=a;i<=a*b;i++)
        {
          if(i%a==0 && i%b==0)
          {
            l=i;
            break;
          }
        }
        System.out.println("L.C.M.:"+l);
    }
}
```

43. Write a program to input two integers and find their Highest Common Factor(H.C.F).

For Example,
INPUT:
Enter 2 integers:
12
8
OUTPUT:
H.C.F. = 4

Ans.

```
import java.util.*;
class Sol43
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,a,b,l=0;
      System.out.println("Enter 2 numbers:");
      a=sc.nextInt();
      b=sc.nextInt();
      for(i=a;i<=a*b;i++)
      {
        if(i%a==0 && i%b==0)
        {
          l=i;
          break;
        }
      }
      System.out.println("HCF.:"+(a*b)/l);
   }
}
```

44. Write a menu driven class to accept a number from the user and check whether it is a Palindrome or a Perfect number.

(a) Palindrome number- (a number is a Palindrome which when read in reverse order is same as read in the right order) Example: 11, 101, 151, etc.

(b) Perfect number- (a number is called Perfect if it is equal to the sum of its factors other than the number itself.) Example: 6=1+2+3

Ans.

```
import java.util.*;
class Sol44
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,n,s=0,ch,d;
      System.out.println("M E N U");
      System.out.println("1. Palindrome Number");
      System.out.println("2. Perfect Number");
      System.out.println("Enter your choice:");
      ch=sc.nextInt();
      System.out.println("Enter a number:");
      n=sc.nextInt();
      switch(ch)
      {
         case 1:
            for(i=n;i>0;i/=10)
            {
               d=i%10;
               s=s*10+d;
            }
            if(s==n)
               System.out.println("Palindrome Number");
            else
               System.out.println("Not a palindrome number");
            break;
        case 2:
            for(i=1;i<n;i++)
            {
               if(n%i==0)
                  s+=i;
            }
            if(s==n)
               System.out.println("Perfect Number");
            else
```

```
            System.out.println("Not a perfect number");
          break;
        default:
          System.out.println("Invalid choice!");
      }
    }
}
```

45. Write a menu driven program to accept a number from the user and check whether it is 'BUZZ' number or to accept any two numbers and print the 'GCD' of them.

(a) A BUZZ number is the number which either ends with 7 or divisible by 7.

(b) GCD (Greatest Common Divisor) of two integers is calculated by continued division method.

Divide the larger number by the smaller; the remainder then divides the previous divisor. The process is repeated till the remainder is zero. The divisor then results the GCD.

Ans.

```
import java.util.*;
class Sol45
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int i,n,r,ch,d;
    System.out.println("M E N U");
    System.out.println("1. Buzz Number");
    System.out.println("2. GCD");
    System.out.println("Enter your choice:");
    ch=sc.nextInt();
    switch(ch)
    {
      case 1:
        System.out.println("Enter a number:");
        n=sc.nextInt();
        if(n%10==7 || n%7==0)
          System.out.println("Buzz Number");
        else
          System.out.println("Not a Buxx number");
        break;
      case 2:
        System.out.println("Enter 2 numbers:");
        n=sc.nextInt();
        d=sc.nextInt();
        do
        {
          r=n%d;
```

```
            if(r!=0)
            {
               n=d;
               d=r;
            }
         }while(r!=0);
         System.out.println("GCD:"+d);
         break;
      default:
         System.out.println("Invalid choice!");
   }
  }
}
```

46. Write a program to find the sum of the following series:

S=1+(1+2)+(1+2+3)+(1+2+3+4)+(1+2+3+4+5)+…+(1+2+3+4+…+10)
(Please note that no nested loop is to be used)

Ans.

```
class Sol46
{
   static void main()
   {
      int i,p=0,s=0;
      for(i=1;i<=10;i++)
      {
         p+=i;
         s+=p;
      }
      System.out.println("Sum:"+s);
   }
}
```

47. Write a program to calculate and print the sum of each of the following series:

(a) Sum (S) = 2 – 4 + 6 – 8 + ……. -20

(b) Sum (S) =x/2 + x/5 + x/8 + x/11 +…+ x/20

Ans.

```
import java.util.*;
class Sol47
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,s=0,ch,x;
      float sum=0;
      System.out.println("M E N U");
```

```
        System.out.println("1. First Series");
        System.out.println("2. Second Series");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
          case 1:
            for(i=2;i<=18;i+=4)
              s+=i-(i+2);
            System.out.println("Sum="+s);
            break;
         case 2:
            System.out.println("Enter the value of x:");
            x=sc.nextInt();
            for(i=2;i<=20;i+=3)
              sum+=(float)x/i;
            System.out.println("Sum:"+sum);
            break;
         default:
            System.out.println("Invalid choice!");
        }
    }
}
```

48. Write a program to find the sum of series, taking the value of 'a' and 'n' from the user.

$$s=\frac{(a+1)}{2}+\frac{(a+3)}{4}+\frac{(a+5)}{6}+\frac{(a+7)}{8}\ldots \text{upto n terms}$$

Ans.

```
import java.util.*;
class Sol48
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a,n,i,f=1;
    float sum=0;
    System.out.println("Enter the value of a:");
    a=sc.nextInt();
    System.out.println("Enter the value of n:");
    n=sc.nextInt();
    for(i=1;i<=n;i++)
    {
      sum+=(float)(a+f)/(f+1);
      f+=2;
```

```
        }
        System.out.println("Sum:"+sum);
    }
}
```

49. Write a program to compute and display the sum of the following series:-

$$s=\frac{(1+2)}{(1*2)}+\frac{(1+2+3)}{(1*2*3)}+\frac{(1+2+3+4)}{(1*2*3*4)}+\frac{(1+2+3+4+5)}{(1*2*3*4*5)}\ldots \text{upto n terms}$$

Ans.

```
import java.util.*;
class Sol49
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a=1,n,i,p=1;
      float sum=0;
      System.out.println("Enter the value of n:");
      n=sc.nextInt();
      for(i=1;i<=n;i++)
      {
         a=a+(i+1);
         p=p*(i+1);
         sum+=(float)a/p;
      }
      System.out.println("Sum:"+sum);
   }
}
```

50. Write a program to generate a triangle or an inverted triangle till n terms based upon the user's choice of triangle to be displayed.

Example 1

Type 1 for a triangle and
Type 2 for an inverted triangle 2
Input: 1
Enter the number of terms 5
1
22
333
4444
55555

Example 2

Type 1 for a triangle and
Type 2 for an inverted triangle 2
Input: 2
Enter the number of terms 6
666666
55555
4444
333
22
1

Ans.

```
import java.util.*;
class Sol50
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,n,j,ch;
      System.out.println("M E N U");
      System.out.println("Type 1 for a triangle and");
      System.out.println("Type 2 for an inverted triangle");
      ch=sc.nextInt();
      System.out.println("Enter the number of terms:");
      n=sc.nextInt();
      switch(ch)
      {
         case 1:
            for(i=1;i<=n;i++)
            {
               for(j=1;j<=i;j++)
               {
                  System.out.print(i);
               }
               System.out.println();
            }
            break;
        case 2:
            for(i=n;i>=1;i--)
            {
               for(j=1;j<=i;j++)
               {
                  System.out.print(i);
               }
               System.out.println();
            }
            break;
        default:
            System.out.println("Invalid choice!");
      }
   }
}
```

51. Write a program to input a number and print whether the number is a special number or not.

(A number is said to be a special number, if the sum of the factorial of the digits of the number is same as the original number).

Example: 145 is a special number, because l! + 4! + 5! = 1+24+120 = 145

(Where ! stands for factorial of the number and the factorial value of a number is the product of all integers from 1 to that number, example 5! = 1*2*3*4*5 = 120).

Ans.

```
import java.util.*;
class Sol51
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,n,j,ch,d,f,s=0;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=n;i>0;i/=10)
      {
         d=i%10;
         f=1;
         for(j=1;j<=d;j++)
            f=f*j;
         s+=f;
      }
      if(s==n)
         System.out.println("Special Number");
      else
         System.out.println("Not a Special Number");
   }
}
```

52. Write a menu driven program to perform the following: (Use switch-case statement)

i. To print the series 0, 3, 7, 15, 24 … n terms (value of n is to be an input by the user)

ii. To find the sum of the series given below:

$$S = \frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \frac{7}{8} \ldots\ldots \frac{19}{20}.$$

Ans.

```
import java.util.*;
class Sol52
{
   static void main()
   {
```

```
        Scanner sc=new Scanner(System.in);
        int i,n,ch;float s=0;
        System.out.println("M E N U");
        System.out.println("1. First Series");
        System.out.println("2. Second Series");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
          case 1:
            System.out.println("Enter the number of terms:");
            n=sc.nextInt();
            for(i=1;i<=n;i++)
            {
              s=i*i-1;
              System.out.print(s+" ");
            }
            break;
          case 2:
            for(i=1;i<=19;i+=2)
            {
              s+=(float)i/(i+1);
            }
            System.out.println("Sum="+s);
            break;
          default:
            System.out.println("Invalid choice!");
        }
      }
    }
```

53. Write a program to input a number and print all its prime factors using prime factorization.

For Example,
INPUT: Enter an integer: 24
OUTPUT: Prime Factors using Prime Factorisation are:
2
2
2
3

Ans.

```
import java.util.*;
class Sol53
{
  static void main()
  {
```

```
        Scanner sc=new Scanner(System.in);
        int n,f=2;
        System.out.println("Enter a number:");
        n=sc.nextInt();
        while(n>1)
        {
          if(n%f==0)
          {
            System.out.println(f);
            n=n/f;
          }
          else
            f++;
        }

    }
}
```

54. Write a program to input a number and check whether it is a Smith number or not. Smith number is such a number, the sum of whose digits equals the sum of the digits of its prime factors.

Smith number is a composite number in which the sum of its digits is equal to the sum of the digits of all its prime factors.

For Example 378 is a Smith Number as the sum of the digits of 378 are : 3+7+8 = 18. The prime factors of 378 are: 2, 3, 3, 3, 7 (sum = 2+3+3+3+7 = 18).

Similarly 22 is a Smith Number as the sum of the digits are : 2+2=4. The prime factors of 22 are: 2 and 11 (Sum = 2+(1+1) = 4

Other Examples include 27, 58, 85, 94, 121, 166, 202, 265, etc.

Ans.

```
import java.util.*;
class Sol54
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int n,f=2,m,t,s1=0,s2=0,d;
    System.out.println("Enter a number:");
    n=sc.nextInt();
    m=n;
    while(n>1)
    {
      if(n%f==0)
      {
```

```
            t=f;
            while(t!=0)
            {
               d=t%10;
               s1+=d;
               t/=10;
            }
            n=n/f;
         }
         else
            f++;
      }
      t=m;
      while(t!=0)
      {
         d=t%10;
         s2+=d;
         t/=10;
      }
      if(s1==s2)
         System.out.println("Smith Number");
      else
         System.out.println("Not a Smith Number");
   }
}
```

55. A special two-digit number is such that when the sum of its digits is added to the product of its digits, the result is equal to the original two-digit number.

Example: Consider the number 59.

Sum of digits = 5 + 9 = 14

Product of its digits = 5 x 9 = 45

Sum of the sum of digits and product of digits = 14 + 45 = 59

Write a program to accept a two-digit number. Add the sum of its digits to the product of its digits.

If the value is equal to the number input, output the message "Special 2-digitnumber" otherwise, output the message "Not a special2-digit number".

Ans.

```
import java.util.*;
class Sol55
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int n,f,l;
```

```
        System.out.println("Enter a number:");
        n=sc.nextInt();
        if(n>=10 && n<=99)
        {
           f=n/10;
           l=n%10;
           if(f+l+f*l==n)
              System.out.println("Special 2-digit number");
           else
              System.out.println("Not a Special 2-digit number");
        }
        else
          System.out.println("Not a 2-digit number");
    }
}
```

56. Using the switch statement, write a menu driven program to calculate the maturity amount of a Bank Deposit.

The user is given the following options:

i. Term Deposit

ii. Recurring Deposit

For option (i) accept principal(P), rate of interest(r) and time period in years(n). Calculate and output the maturity amount(A) receivable using the formula

$$A = P\left[1+\frac{r}{100}\right]^{n}$$

For option (ii) accept Monthly Instalment (P), rate of interest(r) and time period in months(n).

Calculate and output the maturity amount(A) receivable using the formula

$$A = P\times n + P\times \frac{n(n+1)}{2}\times\frac{r}{100}\times\frac{1}{12}$$

For an incorrect option, an appropriate error message should be displayed.

Ans.

```
import java.util.*;
class Sol56
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int ch;
      double p,r,n,A;
      System.out.println("M E N U");
```

```
        System.out.println("1. Term deposit");
        System.out.println("2. Recurring Deposit");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
           case 1:
              System.out.println("Enter the principal:");
              p=sc.nextDouble();
              System.out.println("Enter the rate:");
              r=sc.nextDouble();
              System.out.println("Enter the time period:");
              n=sc.nextDouble();
              A=p*Math.pow(1+r/100,n);
              System.out.println("Amount="+A);
              break;
          case 2:
              System.out.println("Enter the principal:");
              p=sc.nextDouble();
              System.out.println("Enter the rate:");
              r=sc.nextDouble();
              System.out.println("Enter the time period:");
              n=sc.nextDouble();
              A=p*n+p*n*(n+1)/2*r/100*1/12.0;
              System.out.println("Amount="+A);
              break;
          default:
              System.out.println("Invalid choice!");
        }
    }
}
```

57. Use switch statement,write a menu driven program to:

(i) To find and display all the factors of a number input by the user (including 1 and excluding number itself.)

Example:

Sample Input : n=15.

Sample Output : 1,3,5

(ii) To find and display the factorial of a number input by the user(the factorial of a non-negative integer n ,denoted by n!, is the product of all integers less than or equal to n.

Example:

Sample Input : n=5

Sample Output : 5!=1×2×3×4×5=120.

For an incorrect choice, an appropriate error message should be displayed.

Ans.

```
import java.util.*;
class Sol57
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int ch,n,i,f=1;
      System.out.println("M E N U");
      System.out.println("1. Factors");
      System.out.println("2. Factorial");
      System.out.println("Enter your choice:");
      ch=sc.nextInt();
      System.out.println("Enter a number:");
      n=sc.nextInt();
      switch(ch)
      {
         case 1:
            for(i=1;i<n;i++)
            {
               if(n%i==0)
                  System.out.print(i+" ");
            }
            break;
        case 2:
            for(i=1;i<n;i++)
            {
               f=f*i;
            }
            System.out.println("Factorial="+f);
            break;
        default:
            System.out.println("Invalid choice!");
      }
   }
}
```

58. Write a program to accept a number and check and display whether it is a spy number or not.

(A number is spy if the sum its digits equals the product of its digits.)

Example: consider the number 1124 , sum of the digits = 1 + 1 + 2 + 4 = 8

Product of the digits = 1 × 1 × 2 × 4 = 8

Ans.

```
import java.util.*;
class Sol58
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int n,d,p=1,s=0;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      while(n>0)
      {
         d=n%10;
         s+=d;
         p*=d;
         n/=10;
      }
      if(s==p)
         System.out.println("Spy Number");
      else
         System.out.println("Not a spy number");
   }
}
```

59. Using switch statement, write a menu driven program for the following:

(i) To find and display the sum of the series given below:
S = x1 – x2 + x3 0 x4 + x5 ..... – x20
(where x = 2)

(ii) To display the following series:
11 111 1111 11111

For an incorrect option, an appropriate error message should be displayed.

Ans.

```
import java.util.*;
class Sol59
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int ch,n,i,f=0;double s=0;
```

```
        System.out.println("M E N U");
        System.out.println("1. First Series");
        System.out.println("2. Second Series");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
          case 1:
            for(i=1;i<=20;i++)
            {
              if(i%2==1)
                s=s+Math.pow(2,i);
              else
                s=s-Math.pow(2,i);
            }
            System.out.println("Sum="+s);
            break;
          case 2:
            for(i=1;i<=5;i++)
            {
              f=f*10+1;
              System.out.print(f+" ");
            }
            break;
          default:
            System.out.println("Invalid choice!");
        }
      }
    }
```

60. Write a program to input a number and check and print whether it is a Pronic number or not. (Pronic number is the number which is the product of two consecutive integers)

Examples: 12 = 3 × 4
20 = 4 × 5
42 = 6 × 7

Ans.

```
import java.util.*;
class Sol60
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int n,i,f=0;
    System.out.println("Enter a number:");
    n=sc.nextInt();
```

```
        for(i=1;i*(i+1)<=n;i++)
        {
            if(i*(i+1)==n)
            {
                System.out.println("Pronic Number");
                f=1;
            }
        }
        if(f==0)
            System.out.println("Not a pronic number");
    }
}
```

Chapter 3
Class as the Basis of All Computation

Mental Drill

A. Tick (✓) the correct option.

1. Which among the following is not a primitive data type?
   a. int
   b. short
   c. String
   d. Long

   Ans. c. String

2. Name the operator that is used to allocate memory space for an object.
   a. Dot
   b. New
   c. Both a and b
   d. None of these

   Ans. b. New

3. What is the name given to a memory location called in Java?
   a. Variable
   b. Constant
   c. Data Type
   d. None of these

   Ans. a. Variable

4. Which are the data types used in Java?
   a. Primitive data type
   b. Composite data type
   c. Both a and b
   d. None of these

   Ans. c. Both a and b

5. How are the characteristics of an object represented in a class?
   a. Data Members
   b. Member Functions
   c. Access specifiers
   d. None of these

   Ans. a. Data Members

6. Which among the following is used to change the state of an object?
   a. Data Members
   b. Name of the class
   c. Both a and b
   d. None of these

   Ans. a. Data Members

7. How is encapsulation implemented in a program?
   a. Using a class
   b. Using only Functions
   c. Using only Variables
   d. None of these

   Ans. a. Using a class

8. Information passed to a method through arguments is called _____.

a. Message b. Variables

c. Numbers d. Data

Ans. a. Message

9. What among the following is not a component of a message?

a. An object identifier b. A method name

c. Arguments d. Data Members

Ans. d. Data Members

10. How many objects can you create from a class?

a. One b. Two

c. Three d. Any number

Ans. d. Any number

SECTION A

A. Answer the following questions.

1. Mention any two attributes required for class declaration.

Ans. i) The keyword 'class'

ii) The class-name

2. What is a class in Java?

Ans. A class is a mechanism to implement encapsulation. Thus class encloses characteristics represented by data members or variables and behaviour represented by member methods or functions into a single unit. It acts as a blueprint or template for an object.

3. Why is a class called an object factory?

Ans. A class is called an object factory because it acts as a factory for a design or abstraction which produces objects. Each object that is generated from the class will have the same design or abstraction as represented by its data members and member functions.

4. Define Instance Variable. Give an example of the same.

Ans. The data members declared within a class are called instance variables. For example:

```
class Myclass
{
    int a,b;
    void getval(int x, int y)
    {
        a=x;
```

```
            b=y;
        }
    }
```

Here the variables 'a' and 'b' are instance variables of the class 'Myclass'.

5. What does the following mean?

Employee stuff = new Employee( );

Ans. An object named 'stuff' is being created or memory is being allocated of 'Employee' type.

6. How do you declare objects? Show with the help of an example.

Ans. An object is declared using the following general example:

<class-name> <object name>=new <class-constructor>;

For example,

Example ob=new Example();

Here an object named 'ob' is being allocated in the memory of 'Example' type, which is being initialised with the constructor 'Example()'.

7. Write Java statement to create an object mp4 of class digital.

Ans. digital mp4=new digital();

8. What is the difference between an object and a class?

Ans. Difference between class and object:

Class	Object
A class is a template from which objects are created.	An object is an instance of a class.
A class is a group of similar objects.	Object is a real world entity such as pen, laptop, mobile, chair etc.
A class is a logical entity.	An object is a physical entity.

9. What is a variable? How do you initialize a variable?

Ans. Variable is a name given to a memory location in Java.

Initialisation of variables are done during declaration. For example,

int a=5,b=6;

10. What are data types? State its categories.

Ans. Java uses data types to identify data and dictates the method of handling it. It is categorised as Fundamental data type and Composite data type.

11. Why is a class known as a composite data type?

Ans. A composite data type is one which is composed with various primitive data type. A class defined with various primitive data types such as int, double etc. so it is known as a composite data type.

12. Write one difference between primitive data types and composite data types.

Ans.

Fundamental Data Type	Composite Data Type
These are inbuilt data type provided by the Java Language.	These are data types created by the user using fundamental or composite data type or both.
The size of it is fixed.	The size of it depends upon the size of the individual components of it.

13. What does a class encapsulate?

Ans. A class encapsulate characteristics represented by data members and behaviour represented by member functions.

14. Why is a class considered to be an abstraction for a group of objects?

Ans. A class is an abstraction because it describes what is created, whereas an object is created itself. A class acting as an abstraction or blueprint, multiple objects can be created from it.

15. What do you understand by data abstraction? Explain with an example.

Ans. Data Abstraction is the property by virtue of which only the essential details are displayed to the user.

For example, when we ride a bike, we concentrate only driving rather than how the engine is working according to the different switches, lever and steering.

16. Why is an object called an instance of a class?

Ans. An object is called an instance of class as every object created from a class get its own instances of the variables defined in the class.

B. Consider the following code and answer the questions that follow:

```
class academic
{
    int x,y;
    void access()
    {
        int a,b;
        academic student=new academic();
        System.out.println("Object Created");
    }
}
```

a. What is the object name of the class?

b. Name the instance variables used in the class.

c. Write the name of local variables used in the program.

d. Give the type of function being used and its name.

Ans.

a. student
b. x and y
c. a and b
d. Procedural function or pure function.

Name: access()

C. Consider the following code and answer the questions that follow:

```
class vxl
{
    int x,y;
    void init( )
    {
        x=5;
        y=10;
    }
    protected void access( )
    {
        int a=50, b=100;
        vxl vin=new vxl( );
        vin.int( );
        System.out.println("Object created");
        System.out.println("I am X="+vin.x);
        System.out.println("I am Y="+vin.y);
    }
}
```

a. What is the object name of the class vxl?

b. Name the local variables of class.

c. What is the access specifier of method access( )?

d. Write the output of the above program.

Ans.

a. vin

b. a and b

c. protected

d. Output:

Object created

I am X=5

I am Y=10

D. Find the errors in the program given below and rewrite the corrected form:

```
My class
{
    int a;
    int b;
    void display()
    {
        System.out.println(a+“ ”+b);
    }
    static void display2()
    {
        System.out.println(a+“ ”+b);
    }
    public static void main(String args[ ])
    {
        My class ob1=new My class( );
        display1().ob1;
        display2().ob2;
    }
}
```

Ans.

```
class Myclass
{
        int a;
        int b;
        void display1( )
        {
            System.out.println(a+“ ”+b);
        }
        void display2( )
        {
            System.out.println(a+“ ”+b);
        }
        public static void main(String args[ ])
        {
            Myclass ob1=new Myclass( );
            Myclass ob2=new Myclass( );
            ob1.display1();
            ob2.display2();
        }
}
```

SECTION B

Write programs for the following:

1. Write a class with name Employee and basic as its data member, to find the gross pay of an employee for the following allowances and deduction. Use meaningful variables.

Dearness Allowance = 25% of the Basic Pay

House Rent Allowance = 15% of Basic Pay

Provident Fund = 8.33% of Basic Pay

Net Pay = Basic Pay + Dearness Allowance + House Rent Allowance

Gross Pay = Net Pay – Provident Fund

Ans.

```
class Employee
{
   double basic;
   Employee(double b)
   {
      basic=b;
   }

   void calc()
   {
      double pf,gp,np,hra,da;
      da=25/100.0*basic;
      hra=15/100.0*basic;
      pf=8.33/100*basic;
      np=basic+da+hra;
      gp=np-pf;
      System.out.println("Gross Pay="+gp);
   }
}
```

2. Define a class 'Salary' described as below:

Data Members:

Name, Address, Phone, Subject Specialisation, Monthly Salary, Income Tax.

Member methods:

i. To accept the details of a teacher including the monthly salary.

ii. To display the details of the teacher.

iii. To compute the annual Income Tax as 5% of the annual salary above ₹ 1,75,000/-.

Write a main method to create object of the class and call the above member method.

Ans.

```
import java.util.*;
class Salary
{
   String Name, Address,subSpe;
   double mSal,it;
   long phone;
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter your name:");
      Name=sc.nextLine();
      System.out.println("Enter your address:");
      Address=sc.nextLine();
      System.out.println("Enter Subject Specialization:");
      subSpe=sc.nextLine();
      System.out.println("Enter Phone No.:");
      phone=sc.nextLong();
      System.out.println("Enter monthly salary:");
      mSal=sc.nextDouble();
   }
   void display()
   {
      System.out.println("Name:"+Name);
      System.out.println("Address:"+Address);
      System.out.println("Subject Specialization:"+subSpe);
      System.out.println("Phone No.:"+ phone);
      System.out.println("Monthly salary:"+mSal);
   }
   void calc()
   {
      double aSal;
      aSal=12*mSal;
      if(aSal>175000)
         it=5/100.0*(aSal-175000);
      else
         it=0;
   }

   public static void main(String args[])
   {
      Salary ob=new Salary();
      ob.input();
      ob.calc();
```

```
        ob.display();
    }
}
```

3. Define a class 'Student' described as below:

Data members/instance variables: name,age,m1,m2,m3 (marks in 3 subjects), maximum, average Member methods:

i. To accept the details of a student.

ii. To compute the average and the maximum out of three marks.

iii. To display the name, age, marks in three subjects, maximum and average.

Write a main method to create an object of a class and call the above member methods.

Ans.

```
import java.util.*;
class Student
{
    String name;
    int age,m1,m2,m3,max;
    float avg;
    void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your name:");
        name=sc.nextLine();
        System.out.println("Enter marks in 3 subjects:");
        m1=sc.nextInt();
        m2=sc.nextInt();
        m3=sc.nextInt();
        System.out.println("Enter your age:");
        age=sc.nextInt();
    }
    void display()
    {
        System.out.println("Name:"+name);
        System.out.println("Marks:"+m1+","+m2+ "and" +m3);
        System.out.println("Maximum Marks:"+max);
        System.out.println("Average:"+ avg);
    }
    void compute()
    {
        max=Math.max(Math.max(m1,m2),m3);
        avg=(float)(m1+m2+m3)/3;
    }

    public static void main(String args[])
```

```
    {
        Student ob=new Student();
        ob.input();
        ob.compute();
        ob.display();
    }
}
```

4. Define a class Employee having the following description:

Instance variables:

int pan	to store personal account number
String name	to store name
double tax_income	to store annual taxable income
double tax	to store tax that is calculated

Member functions:

input ( )	Store the pan number, name, taxable income
calc( )	Calculate tax for an employee
display ( )	Output details of an employee

Write a program to compute the tax according to the given conditions and display the output as per the given format.

Total Annual Taxable Income	Tax Rate
Upto ₹ 1,00,000	No tax
From 1,00,001 to 1,50,000	10% of the income exceeding ₹ 1,00,000
From 1,50,001 to 2,50,000	₹ 5000 + 20% of the income exceeding ₹ 1,50,000
Above ₹ 2,50,000	₹ 25,000 + 30% of the income exceeding ₹ 2,50,000

Output:

Pan Number	Name	Tax-income	Tax
—	—	—	—

Ans.

```
import java.util.*;

class Employee
{
    String pan, name;
    double tax_income,tax;
    void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your PAN no.:");
        pan=sc.nextLine();
        System.out.println("Enter your name:");
        name=sc.nextLine();
        System.out.println("Enter taxable income:");
```

```
        tax_income=sc.nextDouble();
    }
    void display()
    {
        System.out.println("Pan Number\t\tName\t\tTax-income\t\tTax");
        System.out.println(pan+"\t\t"+name+"\t\t"+tax_income+"\t\t"++ tax);
    }
    void calc()
    {
        if(tax_income<=100000)
            tax=0;
        else if(tax_income>100000 && tax_income<=150000)
            tax=10/100.0*(tax_income-100000);
        else if(tax_income>150001 && tax_income<=250000)
            tax=5000+20/100.0*(tax_income-150000);
        else
            tax=25000+30/100.0*(tax_income-250000);
    }
}
```

5. Define a class called Mobike with the following description:

Instance variables/ Data members:

bno	:	to store the bike's number
phno	:	to store the phone number of the customer
name	:	to store the name of the customer
days	:	to store the number of days the bike is taken on rent
charge	:	to calculate and store the rental charge

Member methods:

void input ()	:	to input and store the detail of the customer
void compute ()	:	to compute the rental charge. The rent for a Mobike is charged on the following basis
First five days	:	₹ 500 per day
Next five days	:	₹ 400 per day
Rest of the days	:	₹ 200 per day
void display ()	:	to display the details in the following format:

Bike No.	Phone No.	Name	No. of days	Charge
- - - - - - -	- - - - - - -	- - - - - -	- - - - - - - -	- - - - - - -

Ans.

```
import java.util.*;
class Mobike
{
    String bno,phno,name;
    int days;
    double charge;
```

```
    void input()
    {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter your bike no.:");
      bno=sc.nextLine();
      System.out.println("Enter your phone no.:");
      phno=sc.nextLine();
      System.out.println("Enter your name:");
      name=sc.nextLine();
      System.out.println("Enter no. of days taken for rent:");
      days=sc.nextInt();
    }
    void display()
    {
      System.out.println("Bike No.\t\tPhone No.\t\tName\t\tNo. of days\t\tCharge");
      System.out.println(bno+"\t\t"+phno+"\t\t"+name+"\t\t"+days+"\t\t"+charge);
    }
    void calc()
    {
      if(days<=5)
        charge=days*500;
      else if(days>5 && days<=10)
        charge=5*500+(days-5)*400;
      else
        charge=5*500+5*400+(days-10)*200;
    }
}
```

6. Write a program with the following specifications:

Class name	:	Student
Data members	:	
name	:	To store the name of a student
hindi	:	To store the marks in hindi subject
english	:	To store the marks in english subject
maths	:	To store the marks in mathematics
computer	:	To store the marks in computer
average	:	To store the avergae of the marks obtained
grade	:	To store the grade depending upon the average.

Member methods:

void accept( )	:	to accept name and marks in the 4 subjects.
void calcavg( )	:	to calculate and store the grade according to the following slabs:

Average marks	**Grade Obtained**
90 and above	A++
Between 75 to 89 (both inclusive)	A

Between 60 to75 (both inclusive) B
Less than 60 C
Write the main method to create the object of the class and call the above method.

Ans.

```
import java.util.*;
class Student
{
   String name;
   int hindi,english,maths,computer;
   float average;
   String  grade;
   void accept()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter your name.:");
      name=sc.nextLine();
      System.out.println("Enter marks in hindi:");
      hindi=sc.nextInt();
      System.out.println("Enter marks in english:");
      english=sc.nextInt();
      System.out.println("Enter marks in maths:");
      maths=sc.nextInt();
      System.out.println("Enter marks in computer:");
      computer=sc.nextInt();
   }
   void calcavg()
   {
      average=(hindi+english+maths+computer)/4f;
      if(average>=90)
         grade="A++";
      else if(average>75 && average<90)
         grade="A";
      else if(average>=60 && average<=75)
         grade="B";
      else
         grade="C";
   }
   public static void main(String args[])
   {
      Student ob=new Student();
      ob.accept();
      ob.calcavg();
   }
}
```

7. Design class called Bank with the following descriptions:

Data members:

name	:	to store the name of the depositor
acno	:	to store the account number
type	:	to store type of the account
bal	:	to store the balance amount in the account

Member functions:

initialise( )	:	to assign the data members with any value.
depo(int a)	:	where a is the amount to be deposited and the variable bal is to be updated.
withdraw(int a)	:	where a is the amount to be withdrawn after checking the balance (Minimum balance should be ₹ 1000) and the variable bal is to be updated.
print( )	:	to print all the details.

Write the main method to create the object of the class and call the above method.

Ans.

```
import java.util.*;
class Bank
{
   String name;
   long acno;
   float bal;
   String  type;
   void initialise()
   {
      name="Saurav Agarwal";
      acno=1001098721;
      bal=10000;
      type="Savings";
   }
   void depo(int a)
   {
      bal+=a;
   }

   void withdraw(int a)
   {
         if(bal-a<1000)
         System.out.println("Minimum balance should be 1000 rupees");
         else
         bal-=a;
   }
```

```
    void print()
    {
      System.out.println("Name:"+name);
      System.out.println("Account No.:"+acno);
      System.out.println("Balance:"+bal);
      System.out.println("Type of Account:"+type);
    }
    public static void main(String args[])
    {
      Bank ob=new Bank();
      Scanner sc=new Scanner(System.in);
      ob.initialise();
      char c;int a;
      System.out.println("Enter (D)eposit/(W)ithdraw:");
      c=sc.next().charAt(0);
      if(c=='D')
      {
        System.out.println("Enter the amount to deposit:");
        a=sc.nextInt();
        ob.depo(a);
      }
      else if(c=='W')
      {
        System.out.println("Enter the amount to withdraw:");
        a=sc.nextInt();
        ob.withdraw(a);
      }
      else
        System.out.println("Invalid input");
      ob.print();
    }
}
```

8. Define a class Bill as described below:

Data members are:

name	:	to store the name of the consumer
consumerno	:	to store the consumer number
unitconsumed	:	to store the unit cosumed
Member methods are	:	
datainput()	:	to read the data of a person
compute()	:	to calculate the bill amount as per criteria.

Units Consumed	**Rate**
Up to 100 units	1.20
More than 100 and up to 200 units	2.20

More than 200 and up to 300 units 3.20
Above 300 units 4.00
Display() – To display the output as per the format:
Consumer Name Consumer No Unit Consumed Bill Amount

Ans.

```
import java.util.*;
class Bill
{
   String name;
   long consumerno;
   int unitconsumed ;

   void datainput()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the name:");
      name=sc.nextLine();
      System.out.println("Enter the consumer no.:");
      consumerno=sc.nextLong();
      System.out.println("Enter the unit consumed:");
      unitconsumed=sc.nextInt();
   }
   float compute()
   {
      float bill=0;
      if(unitconsumed<=100)
         bill=unitconsumed*1.2f;
      else if(unitconsumed>100 && unitconsumed<=200)
         bill=unitconsumed*2.2f;
      else if(unitconsumed>200 && unitconsumed<=300)
         bill=unitconsumed*3.2f;
      else
         bill=unitconsumed*4.0f;
      return bill;
   }

   void Display()
   {
      System.out.println("Consumer Name\t\tConsumer No\t\tUnit Consumed\t\tBill Amount");
      System.out.println(name+"\t\t"+consumerno+"\t\t"+unitconsumed+"\t\t"+compute());
   }
}
```

9. Write a program with the following specifications:

Class : Empl

Data Members:

Emp_No : To store the employee number
Name : To store the name of the employee
Basic : To store the basic salary of an employee
DA : To store the dearness allowance of an employee.
HRA : To store the House Rent Allowance of an employee
TA : To store the Travelling Allowance of an employee
PF : To store the Provident Fund of an employee
Gross : To store the Gross Salary

Member Methods:

get ( ) : To accept Employee No., Name and Basic Salary of the employees
calcu ( ) : To calculate the Gross Salary based on the following condition:

Basic Salary	DA(%)	TA(%)	HRA(%)	PF(%)
>=20,000	53	12	10	8
>=10,000 to <20,000	45	10	12	7.5
< 10,000	40	8	14	7

Gross Salary = (Basic Salary + DA + TA + HRA) – PF
display ( ) : To display the following data in given format:

EMPLOYEE No. NAME GROSS SALARY PF

Write a main method to create the object of the above class and call the above method to calculate and print the Employee No., Name, Gross Salary and PF of an employee.

Ans.

```
import java.util.*;
class Empl
{
   int Emp_No;
   String Name;
   float Basic,DA,HRA,TA,PF,Gross;
   void get()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter your employee no.:");
      Emp_No=sc.nextInt();
      System.out.println("Enter your name:");
      Name=sc.nextLine();
      System.out.println("Enter the basic pay:");
      Basic=sc.nextFloat();
   }
```

```
    void calcu()
    {
      if(Basic>=20000)
      {
        DA=53/100f*Basic;
        TA=12/100f*Basic;
        HRA=10/100f*Basic;
        PF=8/100f*Basic;
      }
      else if(Basic>=10000 && Basic<20000)
      {
        DA=45/100f*Basic;
        TA=10/100f*Basic;
        HRA=12/100f*Basic;
        PF=7.5f/100f*Basic;
      }
      else
      {
        DA=40/100f*Basic;
        TA=8/100f*Basic;
        HRA=14/100f*Basic;
        PF=7/100f*Basic;
      }
      Gross=Basic+DA+TA+HRA+PF-PF;
    }
    void display()
    {
      System.out.println("EMPLOYEE No.\t\tNAME\t\tGROSS SALARY\t\tPF");
      System.out.println(Emp_No+"\t\t"+Name+"\t\t"+Gross+"\t\t"+PF);
    }
    public static void main(String args[])
    {
      Empl ob=new Empl();
      ob.get();
      ob.calcu();
      ob.display();
    }
}
```

10. Define a class called Library with the following description:

Instance variables/data members:

int acc_num	:	stores the accession number of books
String title	:	stores the title of book
String author	:	stores the name of author

Member methods:

void input() : to input and store the accession number, title and author

void compute() : to accept the number of days late, calculate and display the fine charged the rate of ₹ 2 per day

void display() : to display the details in the following format:

Accession number	Title	Author
……	……	……

Write the main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class Library
{
   int acc_num;
   String title,author;
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the accession no.:");
      acc_num=sc.nextInt();
      System.out.println("Enter the title:");
      title=sc.nextLine();
      System.out.println("Enter the author:");
      author=sc.nextLine();
   }
   void compute()
   {
      Scanner sc=new Scanner(System.in);
      int late,fine;
      System.out.println("Enter the no. of days late:");
      late=sc.nextInt();
      fine=late*2;
      System.out.println("Fine:"+fine);
   }
   void display()
   {
      System.out.println("Accession number\t\tTitle\t\tAuthor");
      System.out.println(acc_num+"\t\t"+title+"\t\t"+author);
   }
   public static void main(String args[])
   {
      Library ob=new Library();
```

```
        ob.input();
        ob.compute();
        ob.display();
    }
}
```

11. Define a class called FruitJuice with the following description:

Instance variables/data members:

int product_code	–	stores the product code number
String flavour	–	stores the flavour of the juice.(orange, apple, etc.)
String pack_type	–	stores the type of packaging (tetra-pack, bottle, etc.)
int pack_size	–	stores package size (200ml, 400ml, etc.)
int product_price	–	stores the price of the product

Member Methods:

void input()	–	to input and store the product code, flavour, pack type, pack size and product price
void discount()	–	to reduce the product price by 10
void display()	–	to display the product code, flavour, pack type, pack size and product price

Write the main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class FruitJuice
{
   int product_code,pack_size,product_price;
   String flavour,pack_type;
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the product code:");
      product_code=sc.nextInt();
      System.out.println("Enter the flavour:");
      flavour=sc.nextLine();
      System.out.println("Enter the pack type:");
      pack_type=sc.nextLine();
      System.out.println("Enter the pack size:");
      pack_size=sc.nextInt();
      System.out.println("Enter the product price:");
      product_price=sc.nextInt();
   }
   void discount()
   {
      product_price-=10;
```

```
    }
    void display()
    {
      System.out.println("Product code:"+product_code);
      System.out.println("Flavour:"+flavour);
      System.out.println("Pack type:"+pack_type);
      System.out.println("Pack size:"+pack_size);
      System.out.println("Product price:"+product_price);
    }
    public static void main(String args[])
    {
      FruitJuice ob=new FruitJuice();
      ob.input();
      ob.discount();
      ob.display();
    }
}
```

12. Define a class Book with the following specifications.

Instance variables/data members:

BOOK_NO	:	int type to store the book number
BOOK_TITLE	:	String type to store the title of the book
PRICE	:	float type to store the price per copy

Member Methods:

TOTAL_COST()	:	to calculate the total cost for N number of copies, where N is passed to the function as argument
INPUT()	:	to read BOO_NO, BOOK_TITLE, PRICE
PURCHASE()	:	to ask the user to input the number of copies to be purchased. It

invokes TOTAL_COST() and prints the total cost to be paid by the user. Write the main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class Book
{
  int BOOK_NO;
  String BOOK_TITLE;
  float PRICE;
  void INPUT()
  {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the book no.:");
    BOOK_NO=sc.nextInt();
    System.out.println("Enter the book title:");
```

```
        BOOK_TITLE=sc.nextLine();
        System.out.println("Enter the price:");
        PRICE=sc.nextFloat();
    }
    void TOTAL_COST(int n)
    {
        float tcost;
        tcost=PRICE*n;
        System.out.println("Total Cost:"+tcost);
    }
    void PURCHASE()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no. of copies to purchase:");
        int n=sc.nextInt();
        TOTAL_COST(n);
    }
    public static void main(String args[])
    {
        Book ob=new Book();
        ob.INPUT();
        ob.PURCHASE();
    }
}
```

13. Define a class Flight with the following description:

Instance variables/data members:

fl_no	:	to store the flight number of int type
dest	:	to store the destination of the flight of String type
dist	:	to store the distance of the flight of float type
fuel	:	to store the fuel required by the flight of float type

Member Methods:

i calfuel() : to calculate the value of fuel as per the following criteria Distance Fuel
<=1000 500
>1000 and <=2000 1100
>2000 2200

ii feedinfo() to allow user to enter values for Flight Number, Destination, Distance and call function calfuel() to calculate the quantity of Fuel.

iii showinfo() to allow user to view the content of all the data members.

Write the main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class Flight
{
```

```
    int fl_no;
    String dest;
    float dist,fuel;
    void calfuel()
    {
      if(dist<=1000)
        fuel=500;
      else if(dist>1000 && dist<=2000)
        fuel=1100;
      else
        fuel=2200;
    }

    void feedinfo()
    {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the flight no:");
      fl_no=sc.nextInt();
      System.out.println("Enter the destination:");
      dest=sc.nextLine();
      System.out.println("Enter the distance:");
      dist=sc.nextFloat();
      calfuel();
    }
    void showinfo()
    {
      System.out.println("Flight no:"+fl_no);
      System.out.println("Destination:"+dest);
      System.out.println("Distance:"+dist);
      System.out.println("Fuel:"+fuel);
    }
    public static void main(String args[])
    {
      Flight ob=new Flight();
      ob.feedinfo();
      ob.showinfo();
    }
}
```

14. Define a class hotel in with the following description

Instance variables/data members:

Rno	:	Room No of int type
Name	:	Customer name of String type
Tarrif	:	stores per day charges of float type
NOD	:	no of days integer

Member Methods:

CALC() : to calculate and return Amount as NOD*Tarrif and if the value of NOD*Tarrif is more than 10000 then as 1.05*NOD*Tarrif

Checkin() : to enter the Rno, Name, Tarrif and NOD

Checkout() : to display Rno, Name, Tarrif, NOD and Amount by calling CALC()

Write the main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class hotel
{
   int Rno,NOD;
   String Name;
   float Tarrif;
   float CALC()
   {
      float Amount;
      Amount=NOD*Tarrif;
      if(Amount>10000)
         Amount=Amount*1.05f;
     return Amount;
   }

   void Checkin()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the Room no:");
      Rno=sc.nextInt();
      System.out.println("Enter the Name:");
      Name=sc.nextLine();
      System.out.println("Enter the Tarrif:");
      Tarrif=sc.nextFloat();
      System.out.println("Enter the No. of Days:");
      NOD=sc.nextInt();
   }
   void Checkout()
   {
      System.out.println("Room no:"+Rno);
      System.out.println("Name:"+Name);
      System.out.println("Tarrif:"+Tarrif);
      System.out.println("No. of days:"+NOD);
      System.out.println("Amount:"+CALC());
   }
   public static void main(String args[])
   {
```

```
        hotel ob=new hotel();
        ob.Checkin();
        ob.Checkout();
    }
}
```

15. Define a class Telephone having the following description:

Instance Variables / Data Members:

int prv, pre	-	to store the previous and present meter reading
int call	-	to store the calls made (i.e. pre – prv)
String name	-	to store name of the customer
double amt	-	to store the amount
double total	-	to store the total amount to be paid

Member Methods:

void input ( )	-	to input the previous reading, present reading and name of the customer
void cal ( )	-	to calculate the amount and total amount to be paid
void display ( )	-	to display the name of the customer, calls made, amount and total amount to be paid in the following format:

Name	Calls Made	Amount	Total Amount
.....	.....	.....	.....

Write a program to compute the monthly bill to be paid according to the given conditions:

Calls made	Rate
Upto 100 calls	No Charge
For the next 100 calls	90 paise per call
For the next 200 calls	80 paise per call
More than 400 calls	70 paise per call

However every customer has to pay ₹ 180 per month as monthly rent for availing the service.

Ans.

```
import java.util.*;
class Telephone
{
   int prv, pre,call;
   String name;
   double amt,total;
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the previous meter reading:");
      prv=sc.nextInt();
      System.out.println("Enter the present meter reading:");
      pre=sc.nextInt();
```

```
    System.out.println("Enter the name:");
    name=sc.nextLine();
  }
  void cal()
  {
    call=pre-prv;
    if(call<=100)
      amt=0;
    else if(call>100 && call<=200)
      amt=0*100+(call-100)*0.90;
    else if(call>200 && call<=400)
      amt=0*100+100*0.90+(call-200)*0.80;
    else
      amt=0*100+100*0.90+200*0.80+(call-400)*0.70;
    total=amt+180;
  }


  void display()
  {
    System.out.println("Name\t\tCalls Made\t\tAmount\t\tTotal Amount");
    System.out.println(name+"\t\t"+call+"\t\t"+amt+"\t\t"+total);
  }
  public static void main(String args[])
  {
    Telephone ob=new Telephone();
    ob.input();
    ob.cal();
    ob.display();
  }
}
```

16. Define a class named movieMagic with the following description:

Instance variables/data members:

int year	–	to store the year of release of a movie
String title	–	to store the title of the movie.
float rating	–	to store the popularity rating of the movie.

(minimum rating = 0.0 and maximum rating = 5.0)

Member Methods:

(i) void accept()	-	To input and store year, title and rating.
(ii) void display()	-	To display the title of a movie and a message based on the rating as per the table below.

Rating	Message to be displayed
0.0 to 2.0	Flop
2.1 to 3.4	Semi-hit
3.5 to 4.5	Hit
4.6 to 5.0	Super Hit

Write a main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class movieMagic
{
   int year;
   String title;
   float rating;
   void accept()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the year:");
      year=sc.nextInt();
      System.out.println("Enter the title:");
      title=sc.nextLine();
      System.out.println("Enter the rating:");
      rating=sc.nextFloat();
   }
   void display()
   {
         System.out.println("Title:"+title);
         if(rating>=0.0f && rating<=2.0f)
         System.out.println("Flop");
         else if(rating>=2.1f && rating<=3.4f)
         System.out.println("Semi Hit");
         else if(rating>=3.5f && rating<=4.5f)
         System.out.println("Hit");
         else if(rating>=4.6f && rating<=5.0f)
         System.out.println("Super Hit");
   }

   public static void main(String args[])
   {
      movieMagic ob=new movieMagic();
      ob.accept();
      ob.display();
   }
}
```

17. Define a class ParkingLot with the following description:

Instance variables/data members:

int vno	–	To store the vehicle number
int hours	–	To store the number of hours the vehicle is parked in the parking lot
double bill	–	To store the bill amount

Member methods:

void input()	–	To input and store vno and hours
void calculate()	–	To compute the parking charge at the rate of ` 3 for the first hour or part thereof, and ` 1.50 for each additional hour or part thereof.
void display()	–	To display the detail

Write a main method to create an object of the class and call the above methods

Ans.

```
import java.util.*;
class ParkingLot
{
   int vno,hours;
   double bill;
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the vehicle number:");
      vno=sc.nextInt();
      System.out.println("Enter the number of hours:");
      hours=sc.nextInt();
   }
   void calculate()
   {
      if(hours<=1)
         bill=3*hours;
      else
         bill=3*1+(hours-1)*1.50;
   }
   void display()
   {
      System.out.println("Vehicle number:"+vno);
      System.out.println("Number of hours:"+hours);
      System.out.println("Bill:"+bill);
   }
   public static void main(String args[])
   {
      ParkingLot ob=new ParkingLot();
      ob.input();
```

```
        ob.calculate();
        ob.display();
    }
}
```

18. Define a class named BookFair with the following description:

Instance variables/Data members:

String Bname – stores the name of the book.

double price – stores the price of the book.

Member Methods:

(i) void Input() – To input and store the name and the price of the book.

(ii) void calculate() – To calculate the price after discount. Discount is calculated based on the following criteria:

PRICE	DISCOUNT
Less than or equal to ₹ 1000	2% of price
More than ₹ 1000 and less than or equal or ₹ 3000	10% of price
More than ₹ 3000	15% of price

(iii) void display() – To display the name and price of the book after discount.

Write a main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class BookFair
{
    String Bname;
    double price;
    void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the name of the book:");
        Bname=sc.nextLine();
        System.out.println("Enter the price of the book:");
        price=sc.nextDouble();
    }
    void calculate()
    {
        double dis=0;
        if(price<=1000)
            dis=2/100.0*price;
        else if(price>1000 && price<=3000)
            dis=10/100.0*price;
        else
            dis=15/100.0*price;
```

```
        price=price-dis;
    }
    void display()
    {
        System.out.println("Name:"+Bname);
        System.out.println("Price:"+price);
    }
    public static void main(String args[])
    {
        BookFair ob=new BookFair();
        ob.input();
        ob.calculate();
        ob.display();
    }
}
```

19. Define a class Electric Bill with the following specifications:

class: ElectricBill

Instance Variable/ data member:

String n	–	to store the name of the customer
int units	–	to store the number of units consumed
double bill	–	to store the amount to paid

Member methods:

• void accept()	–	to accept the name of the customer and number of units consumed
• void calculate()	–	to calculate the bill as per the following tariff :
Number of units	–	Rate per unit
First 100 units	–	₹ 2.00
Next 200 units	–	₹ 3.00
Above 300 units	–	₹ 5.00

A surcharge of 2.5% charged if the number of units consumed is above 300 units.

• void print()	–	To print the details as follows :

Name of the customer ……

Number of units consumed ……

Bill amount …….

Write a main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class ElectricBill
{
    String n;
    int units;
    double bill;
    void accept()
```

```
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the name of the customer:");
        n=sc.nextLine();
        System.out.println("Enter the units consumed:");
        units=sc.nextInt();
    }
    void calculate()
    {
        double rate=0;
        if(units<=100)
            rate=2*units;
        else if(units>100 && units<=300)
            rate=100*2+(units-100)*3;
         else
            rate=100*2+200*3+(units-300)*5;

        if(units>300)
            bill=rate+2.5/100*rate;
        else
            bill=rate;
    }
    void print()
    {
        System.out.println("Name of the customer:"+n);
        System.out.println("Number of units consumed:"+units);
        System.out.println("Bill amount:"+bill);
    }
    public static void main(String args[])
    {
        ElectricBill ob=new ElectricBill();
        ob.accept();
        ob.calculate();
        ob.print();
    }
}
```

20. Design a class RailwayTicket with the following description:

Instance variables/data members:

String name : To store the name of the customer
String coach : To store the type of coach customer wants to travel
long mobno : To store customer's mobile number

int amt : To store basic amount of ticket
int totalamt : To store the amount to be paid after updating the original amount

Member methods:

- void accept() - To take input for name, coach, mobile number and amount
- void update() - To update the amount as per the coach selected

(extra amount to be added in the amount as follows)

Type of Coaches	Amount
First_AC	700
Second_AC	500
Third_AC	250
Sleeper	None

- void display() - To display all details of a customer such as name, coach, total amount and mobile number.

Write a main method to create an object of the class and call the above member methods.

Ans.

```
import java.util.*;
class RailwayTicket
{
   String name, coach;
   long mobno;
   int amt,totalamt;
   void accept()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the name:");
      name=sc.nextLine();
      System.out.println("Enter the type of coach:");
      coach=sc.nextLine();
      System.out.println("Enter the mobile number:");
      mobno=sc.nextLong();
      System.out.println("Enter the amount:");
      amt=sc.nextInt();
   }
   void update()
   {
      double rate=0;
      if(coach.equals("First_AC"))
         totalamt=amt+700;
      else if(coach.equals("Second_AC"))
         totalamt=amt+500;
```

```
        else if(coach.equals("Third_AC"))
            totalamt=amt+250;
        else if(coach.equals("Sleeper"))
            totalamt=amt;

    }
    void print()
    {
        System.out.println("Name of the customer:"+name);
        System.out.println("Type of coach:"+coach);
        System.out.println("Total amount:"+totalamt);
        System.out.println("Mobile Number:"+mobno);
    }
    public static void main(String args[])
    {
        RailwayTicket ob=new RailwayTicket();
        ob.accept();
        ob.update();
        ob.print();
    }
}
```

Chapter 4
User-Defined Methods

Mental Drill

A. Tick (√) the correct option.

1. Once a function is defined, it can be invoked repeatedly. What is this feature called?

   a. Interface
   b. Reusability
   c. Restructuring
   d. All of these

   Ans. b. Reusability

2. If a function does not return any value, its return type should be:

   a. int
   b. no-return
   c. void
   d. empty

   Ans. c. void

3. A function that computes a value and returned is called:

   a. Computational function
   b. Manipulative function
   c. Procedural function
   d. None of these

   Ans. a. Computational function

4. A type of parameter that are used to identify what data is to be passed to a function is called:

   a. Formal parameter
   b. Actual parameter
   c. Both a and b
   d. None of these

   Ans. a. Formal parameter

5. The parameter list in a function definition is called:

   a. Function prototype
   b. Function signature
   c. Both a and b
   d. None of these

   Ans. b. Function signature

6. The first line of the function definition is called:

   a. Function prototype
   b. Function signature
   c. Both a and b
   d. None of these

   Ans. a. Function prototype

7. The number of values that a function can return is:

   a. 1
   b. 2
   c. 3
   d. 4

   Ans. a. 1

8. A local variable in a function has its scope:

   a. Limited within the function

   b. Can be accessed anywhere within the same class

   c. No limitation at all

   d. None of these

Ans. a. Limited within the function

9. Which among the following is a valid name for a function?

   a. function b. 2function

   c. fun in action d. fun#

Ans. a. function

10. Which among the following is not a valid access specifier?

   a. public b. private

   c. partially d. protected

Ans. c. partially

B. Fill in the blanks.

1. In Java functions are known as **methods**.
2. Methods are contained in **class**.
3. The function name and the parameter list together is known as **function signature**.
4. The access specifier, return type and the function signature together is known as **prototype**.
5. The arguments of the function given in function prototype are called **formal arguments**.
6. The arguments of the function given in statement that calls the function are called **actual arguments**.
7. If a function does not return any value, the returning type in the function prototype will be **void**.
8. When a function is called by **value**, the values of actual parameters are copied into separate memory locations as allocated by the formal parameters.
9. Impure functions uses call by **reference**.
10. One of the practical implementation of polymorphism is **overloading**.

SECTION A

A. Answer the following questions.

1. What is a method?

Ans. A Java method is a collection of statements that are grouped together to perform an operation.

2. Write two advantages of using functions in a program.

Ans. Reduces complexity and Reusability.

3. Explain the function of the 'return' statement.

Ans. The 'return' statement is used to return a value from a method, before exiting from the function.

4. If a function contains several return statements, how many of them will be executed?

Ans. The first one.

5. Name the keyword that causes the control to transfer back to the method call.

Ans. return

6. What is the role of the keyword 'void' in declaring functions?

Ans. The void keyword ensures that a method do not return any value.

7. Classify functions depending upon the value it returns.

Ans. Computational function, Procedural functions and Manipulative functions

8. Differentiate between Formal Parameter and Actual Parameter.

Ans. Difference:

Formal Parameters	Actual Parameters
Parameter list in the function prototype.	Parameter list in the function invocation.
They are only variables.	They may be both variables or constants.

9. State the difference between function prototype and function signature.

Ans. Function prototype is the first line of the function definition that consist of return type function name and the parameter list. Function signature on the other hand only specifies the parameter list.

10. How are functions called? How does it return a value?

Ans. Functions are called using the name of the function, they may be called using either:

- Call by Value
- Call by reference

The return statement is used to return a value from a function.

11. State the difference between Call by Value and Call by Reference.

Ans.

Difference between Call by Value and Call by Reference	
Call by Value	**Call by Reference**
Using this technique a copy of the values in the actual parameter is made in the formal parameters.	Using this technique a reference to the values in the actual parameter is made by the formal parameters.

Any changes made to the formal parameters is not reflected back in the actual parameters.	Any changes made to the formal parameters is reflected back in the actual parameters.
Usually primitive data type is used during call by value.	Usually arrays and objects are used during call by reference.

12. How are the following passed?

(i) Primitive types (ii) Reference types

Ans. (i) Call by value

(ii) Call by reference

13. Differentiate between pure and impure function.

Ans.

Difference between Pure and Impure Functions	
Pure Functions	**Impure Functions**
Pure functions are such functions which do not change the state of an object.	Impure functions are such functions which changes the state of an object.
It doesn't have any side effects as the state of the object is not changed, rather only accessed.	It does have side effects as the state of the object is changed, and therefore one should be careful using it.

14. Explain function overloading with an example.

Ans.

Example of function overloading:

```
class Overload
{
   static void num_calc(int num,char ch)
   {
      if(ch=='s')
         System.out.println("Square:"+(num*num));
      else
         System.out.println("Square:"+(num*num*num));
   }
   static void num_calc(int a,int b,char ch)
   {
      if(ch=='p')
         System.out.println("Product:"+(a*b));
      else
         System.out.println("Sum:"+(a+b));
   }
   static void num_calc(String s1,String s2)
   {
```

```
        if(s1.equalsIgnoreCase(s2))
            System.out.println("Equal");
        else
            System.out.println("Not Equal");
    }
}
```

15. Which OOP principle implements function overloading?

Ans. Polymorphism

16. When there are multiple definitions with the same function name, what makes them different from each other?

Ans. Function signature

17. What are the different access specifiers available in Java?

Ans. default, public, private and protected

18. What is the use of main() method?

Ans. The main() method, in general, is from where a program execution begins in Core Java.

19. How are static methods of one class called by methods in other classes?

Ans. The function invocation should be preceded with the name of the class followed by the dot operator.

B. Write function prototypes for the following:

1. Private access method sum which accepts three int type variables as parameters and return a float type.

Ans. private float sum(int a,int b,int c)

2. Default access method "length" which accepts a String type variable as parameter and returns an int type.

Ans. int length(String s)

3. Public access method "increment" which accepts an object of Myclass type as parameter and does not return any value.

Ans. public void increment(Myclass ob)

4. Protected access method largest which accepts a float type, int type and double type data as parameters and have a byte type as return type.

Ans. protected byte largest(float a, int b, double c)

5. Public access static method calculate which accepts a byte and int type data type as parameters and return float type.

Ans. public static float calculate(byte b, int d)

6. Write the function prototype for the function "sum" that takes an integer variable (x) as its argument and returns a value of float data type.

Ans. float sum(int x)

7. Write the prototype of a function which takes in 2 integers and 1 String arguments and returns a value which is either 'true' or false'.

Ans. boolean function(int a,int b,String c)

C. Answer as directed:

1. In the program given below:

```
class MyClass
{
    static int x = 7;
    int y = 2;
    public static void main(String args[ ])
    {
        MyClass obj = new MyClass();
        System.out.println(x);
        obj.sampleMethod(5);
        int a= 6;
        System.out.println(a);
    }
        void sampleMethod(int n)
    {
        System.out.println(n);
        System.out.println(y);
    }
}
```

State the name and value of the:

(i) method argument or argument variable.

(ii) class variable.

(iii) local variable.

(iv) instance variable.

Ans.

(i) main() =String args[] value=none
sampleMethod=int n value=5

(ii) x value=7

(iii) a value=6

(iv) y value=2

SECTION B

Write programs for the following:

1. Create a method which accepts two int type variable a and b as parameter and evaluate the following expression:

$\frac{4.52a+b^2}{a-b}$ **and return it.**

Ans.

```
class Sol1
{
   static double method(int a,int b)
   {
      double c;
      c=(4.2*a+b*b)/(a-b);
      return c;
   }
}
```

2. Create a function which accepts an integer as parameter and return true if it is a prime number otherwise return false. In the main() method input an integer and using the above method check whether it is a prime number or not.

Ans.

```
import java.util.*;
class Sol2
{
   static boolean isPrime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }

   static void main()
   {
      Scanner sc=new Scanner(System.in);
```

```
        int n;
        System.out.println("Enter a number:");
        n=sc.nextInt();
        if(isPrime(n)==true)
            System.out.println("Prime Number");
        else
            System.out.println("Not a Prime Number");
    }
}
```

3. Create a function which accepts an integer as parameter and return true if it is a perfect number otherwise return false. In the main() method input two integers and check whether both the numbers are perfect numbers or not.

Ans.

```
import java.util.*;
class Sol3
{
    static boolean isPerfect(int n)
    {
        int i,s=0;
        for(i=1;i<n;i++)
        {
            if(n%i==0)
                s+=i;
        }
        if(s==n)
            return true;
        else
            return false;
    }

    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a,b;
        System.out.println("Enter 2 numbers:");
        a=sc.nextInt();
        b=sc.nextInt();
        if(isPerfect(a)==true && isPerfect(b)==true)
            System.out.println("Both are prefect numbers");
        else
            System.out.println("Both are not perfect numbers");
    }
}
```

4. Create a function which accepts an integer as parameter and return the sum of the square of its digits. In the main() method display all three digit Armstrong numbers.

Ans.

```
class Sol4
{
   static int sum(int n)
   {
      int i,s=0,d;
      for(i=n;i>0;i=i/10)
      {
         d=i%10;
         s+=d*d*d;
      }
      return s;
   }

   static void main()
   {
      int i;
      for(i=100;i<=999;i++)
      {
         if(sum(i)==i)
            System.out.println(i);
      }
   }
}
```

5. Create a function which accepts an integer as parameter and return true if it is a palindrome number or not. In the main() method input 10 integers and print the largest palindrome number if any.

Ans.

```
import java.util.*;
class Sol5
{
   static boolean isPal(int n)
   {
      int i,r=0,d;
      for(i=n;i>0;i=i/10)
      {
         d=i%10;
         r=r*10+d;
      }
      if(r==n)
         return true;
```

```
        else
            return false;
    }

    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,max=0,n;
        System.out.println("Enter 10 numbers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            if(isPal(n)==true)
            {
                if(max==0)
                    max=n;
                if(n>max)
                    max=n;
            }
        }
        if(max>0)
            System.out.println("Largest Palindrome Number:"+max);
        else
            System.out.println("No palindrome number present");
    }
}
```

6. Create a function which accepts an integer as parameter and return the sum of its digits. Create another function to input 10 integers and find the sum of the digits of each number.

Ans.

```
import java.util.*;
class Sol6
{
    static int sum(int n)
    {
        int i,s=0,d;
        for(i=n;i>0;i=i/10)
        {
            d=i%10;
            s=s+d;
        }
        return s;
    }
    static void main()
    {
```

```
        Scanner sc=new Scanner(System.in);
        int i,s,n;
        System.out.println("Enter 10 numbers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            s=sum(n);
            System.out.println("Sum of the digits of"+n+"is"+s);
        }
    }
}
```

7. Create a function which accepts an integer as parameter and return the largest digit. Create another function to input 10 integers and find the sum of the largest digit of each number.

Ans.

```
import java.util.*;
class Sol7
{
    static int largest(int n)
    {
        int i,max=0,d;
        for(i=n;i>0;i=i/10)
        {
            d=i%10;
            if(d>max)
                max=d;
        }
        return max;
    }

    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,s=0,n,max;
        System.out.println("Enter 10 numbers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            max=largest(n);
            s+=max;
        }
        System.out.println("Sum ="+s);
    }
}
```

8. Create a method which accepts temperature in Celsius and return its Fahrenheit equivalent. Create another method which accepts temperature in Fahrenheit and return its Celsius equivalent. Also create a method to invoke the above methods.

Ans.

```
import java.util.*;
class Sol8
{
   static double CelToFar(double C)
   {
      double F=9*C/5+32;
      return F;
   }
   static double FarToCel(double F)
   {
      double C=(F-32)*5/9;
      return C;
   }
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      double C,F;
      System.out.println("Enter temperature in Celcius:");
      C=sc.nextDouble();
      F=CelToFar(C);
      System.out.println("Temperature in Farenheit:"+F);
      System.out.println("Enter temperature in Farenheit:");
      F=sc.nextDouble();
      C=FarToCel(F);
      System.out.println("Temperature in Celcius:"+C);
   }
}
```

9. Create a class with the following methods:

a. int sum(int n), which finds the sum of the digits in n and returns it.

b. void call() to input an integer using scanner and find the sum of its digits using the above method.

Ans.

```
import java.util.*;
class Sol9
{
   static int sum(int n)
   {
      int i,d,s=0;
      for(i=n;i>0;i=i/10)
```

```
    {
      d=i%10;
      s+=d;
    }
    return s;
  }

  static void call()
  {
    Scanner sc=new Scanner(System.in);
    int n,s;
    System.out.println("Enter a number:");
    n=sc.nextInt();
    s=sum(n);
    System.out.println("Sum:"+s);
  }
}
```

10. Create a class with the following methods:

a. int sum(int n), which finds the sum of the digits in n and returns it.

b. void call() to input 10 integers using scanner and find the sum of its digits of each integer value entered by the user using the above method.

Ans.

```
import java.util.*;
class Sol10
{
  static int sum(int n)
  {
    int i,d,s=0;
    for(i=n;i>0;i=i/10)
    {
      d=i%10;
      s+=d;
    }
    return s;
  }

  static void call()
  {
    Scanner sc=new Scanner(System.in);
    int n,s,i;
    System.out.println("Enter 10 numbers:");
    for(i=1;i<=10;i++)
    {
      n=sc.nextInt();
```

```
            s=sum(n);
            System.out.println("Sum of the digits in:"+n+ "is"+s);
        }
    }
}
```

11. Create a class with the following methods:

a. boolean prime(int n), which returns true if n is prime otherwise returns false.

b. void call() to input 10 integers using scanner and find the sum of those integers which are prime numbers using the above method.

Ans.

```
import java.util.*;
class Sol11
{
    static boolean prime(int n)
    {
        int i,c=0;
        for(i=1;i<=n;i++)
        {
            if(n%i==0)
                c++;
        }
        if(c==2)
            return true;
        else
            return false;
    }

    static void call()
    {
        Scanner sc=new Scanner(System.in);
        int n,i,s=0;
        System.out.println("Enter 10 numbers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            if(prime(n)==true)
                s+=n;
        }
        System.out.println("Sum of Prime Numbers:"+s);
    }
}
```

12. Create a class with the following methods:

a. boolean is Armstrong(int n), which returns true if n is Armstrong number otherwise returns false.
Armstrong numbers are those numbers, whose sum of the cube of its digits is equal to the number.
Eg. 153 = $1^3 + 5^3 + 3^3$

b. void call() to input 10 integers using scanner and find largest Armstrong number if any.

Ans.

```
import java.util.*;
class Sol12
{
  static boolean isArmstrong(int n)
  {
    int i,s=0,d;
    for(i=n;i>0;i=i/10)
    {
      d=i%10;
      s+=d*d*d;
    }
    if(s==n)
      return true;
    else
      return false;
  }

  static void call()
  {
    Scanner sc=new Scanner(System.in);
    int i,n,max=0;
    System.out.println("Enter 10 numbers:");
    for(i=1;i<=10;i++)
    {
      n=sc.nextInt();
      if(isArmstrong(i)==true)
      {
        if(max==0)
          max=n;
        if(n>max)
          max=n;
      }
    }
    if(max>0)
        System.out.println("Largest armstrong number:"+max);
    else
```

```
            System.out.println("No armstrong number present");
        }
    }
```

13. Create a class with the following methods:

a. boolean prime(int n), which returns true if n is prime otherwise returns false.

b. void call() to input 2 integers using scanner and print only the prime numbers between the given integers.

Ans.

```
import java.util.*;
class Sol13
{
    static boolean prime(int n)
    {
        int i,c=0;
        for(i=1;i<=n;i++)
        {
            if(n%i==0)
                c++;
        }
        if(c==2)
            return true;
        else
            return false;
    }

    static void call()
    {
        Scanner sc=new Scanner(System.in);
        int i,a,b,s,l;
        System.out.println("Enter 2 numbers:");
        a=sc.nextInt();
        b=sc.nextInt();
        if(a>b)
        {
            l=a;
            s=b;
        }
        else
        {
            l=b;
            s=a;
        }
        for(i=s+1;i<l;i++)
        {
```

```
            if(prime(i)==true)
               System.out.println("Prime number:"+i);
         }
      }
   }
```

14. Create a class with the following methods:

a. boolean prime(int n), which returns true if n is prime otherwise returns false.
b. int sum(int n), which finds the sum of the digits in n and returns it.
c. void call() to display all such three digit numbers, whose sum of the digits is a prime number.

Ans.

```
class Sol14
{
   static boolean prime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }
   static int sum(int n)
   {
      int i,d,s=0;
      for(i=n;i>0;i=i/10)
      {
         d=i%10;
         s+=d;
      }
      return s;
   }
   static void call()
   {
      int i,s;
      for(i=100;i<=999;i++)
      {
         s=sum(i);
         if(prime(s)==true)
            System.out.println("Prime number:"+i);
      }
   }
}
```

15. Create a class with the following methods:

a. boolean prime(int n), which returns true if n is prime otherwise returns false.

b. boolean palindrome(int n), which returns true if n is palindrome otherwise returns false.

c. void call() to display all three digit pal-prime numbers. Pal-prime numbers are those numbers which are both palindrome as well as prime.

Ans.

```
class Sol15
{
   static boolean prime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }
   static boolean palindrome(int n)
   {
      int i,r=0,d;
      for(i=n;i>0;i=i/10)
      {
         d=i%10;
         r=r*10+d;
      }
      if(r==n)
         return true;
      else
         return false;
   }
   static void call()
   {
      int i,s;
      for(i=100;i<=999;i++)
      {
         if(prime(i)==true && palindrome(i)==true)
            System.out.println("Pal-Prime number:"+i);
      }
   }
}
```

16. Create a class with the following methods:

a. boolean prime(int n), which returns true if n is prime otherwise returns false.

b. void call() to display all two digit twin-prime numbers. Twin-prime numbers are those pair of numbers which are both prime as well as whose difference is 2. For example, 11 and 13, 17 and 19, etc.

Ans.

```
class Sol16
{
  static boolean prime(int n)
  {
    int i,c=0;
    for(i=1;i<=n;i++)
    {
      if(n%i==0)
        c++;
    }
    if(c==2)
    return true;
    else
    return false;
    }

    static void call()
    {
        int i,s;
        for(i=10;i<=98;i++)
    {
        if(prime(i)==true && prime(i+2)==true)
        System.out.println("Twin-Prime number:"+i+" "+(i+2));
    }
  }
}
```

17. Create a class with the following methods:

a. boolean prime(int n), which returns true if n is prime otherwise returns false.

b. int sumPrime(int n), which returns the sum of prime digits present in integer n.

c. void call() to input an integer and check whether the sum of prime digits is also a prime number or not.

Ans.

```
import java.util.*;
class Sol17
{
  static boolean prime(int n)
  {
```

```
    int i,c=0;
    for(i=1;i<=n;i++)
    {
      if(n%i==0)
        c++;
    }
    if(c==2)
      return true;
    else
      return false;
  }
  static int sumPrime(int n)
  {
    int i,s=0,d;
    for(i=n;i>0;i=i/10)
    {
      d=i%10;
      if(prime(d)==true)
        s=s+d;
    }
    return s;
  }
  static void call()
  {
    int i,s,n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number:");
    n=sc.nextInt();
    s=sumPrime(n);
    if(prime(s)==true)
      System.out.println("Sum of the prime digits is also a prime number");
    else
      System.out.println("Sum of the prime digits is not a prime number");
  }
}
```

18. Create a class with the following methods:

a. int largest(int a,int b), which returns largest among the two integers a and b and return it.

b. void call() to input 10 integers and using the above function find the largest among the 10 integers.

Ans.

```
import java.util.*;
class Sol18
{
```

```
    static int largest(int a,int b)
    {
      if(a>b)
        return a;
      else
        return b;
    }

    static void call()
    {
      int i,max=0,n;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter 10 numbers:");
      for(i=1;i<=10;i++)
      {
        n=sc.nextInt();
        if(max==0)
          max=n;
        else
          max=largest(max,n);
      }
      System.out.println("Largest:"+max);
    }
}
```

19. Create a class with the following methods:

a. int LCM(int a,int b), which returns the least common multiple(LCM) among the two integers a and b and return it.

b. void call() to input 10 integers and using the above function find the LCM among the 10 intgers.

Ans.

```
import java.util.*;
class Sol19
{
  static int LCM(int a,int b)
  {
    int i;
    for(i=a;i<=a*b;i++)
    {
      if(i%a==0 && i%b==0)
        break;
    }
    return i;
  }
```

```
    static void call()
    {
      int i,lcm=0,n;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter 10 numbers:");
      for(i=1;i<=10;i++)
      {
        n=sc.nextInt();
        if(lcm==0)
          lcm=n;
        else
          lcm=LCM(lcm,n);
      }
      System.out.println("Largest:"+lcm);
    }
}
```

20. Create a class with the following methods:

a. int HCF(int a,int b), which returns the highest common factor(HCF) among the two integers a and b and return it.

b. void call() to input 10 integers and using the above function find the HCF among the 10 intgers.

Ans.

```
import java.util.*;
class Sol20
{
  static int HCF(int a,int b)
  {
    int i;
    for(i=a;i<=a*b;i++)
    {
      if(i%a==0 && i%b==0)
        break;
    }
    return (a*b)/i;
  }

  static void call()
  {
    int i,hcf=0,n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter 10 numbers:");
    for(i=1;i<=10;i++)
    {
      n=sc.nextInt();
```

```
            if(hcf==0)
               hcf=n;
            else
               hcf=HCF(hcf,n);
         }
         System.out.println("Largest:"+hcf);
      }
   }
```

21. Create a class called GeneratePrime which will be used to generate n number of prime numbers. The class should have the following methods:

(i) Method called isPrime() which accepts an integer as a parameter and return true if it is a prime number otherwise return false.

(ii) Method called display() which accepts an integer n as Scanner input and display the first n prime number by calling the above function.

Ans.

```
import java.util.*;
class GeneratePrime
{
   static boolean isPrime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }

   static void display()
   {
      int i=0,n,p=2;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a number:");
      n=sc.nextInt();
      while(i<n)
      {
         if(isPrime(p)==true)
         {
           System.out.println(p);
           i++;
```

```
            }
            p++;
        }
    }
}
```

22. Create a class with the following Member Functions:

(i) Method called isPrime() which accepts an integer as a parameter and return true if it is a prime number otherwise return false.

(ii) Method called sumOfPrimeDigits() which accepts an integer as a parameter and return the sum of prime digits only.

(iii) Method called check() which accepts 10 integers and checks whether the sum of the prime numbers is also a prime number or not.

Ans.

```
import java.util.*;
class Sol22
{
    static boolean isPrime(int n)
    {
        int i,c=0;
        for(i=1;i<=n;i++)
        {
            if(n%i==0)
                c++;
        }
        if(c==2)
            return true;
        else
            return false;
    }
    static int sumOfPrimeDigits(int n)
    {
        int i,s=0,d;
        for(i=n;i>0;i=i/10)
        {
            d=i%10;
            if(isPrime(d)==true)
                s=s+d;
        }
        return s;
    }
    static void check()
    {
        int i,s=0,n;
        Scanner sc=new Scanner(System.in);
```

```
        System.out.println("Enter 10 numbers:");
        for(i=1;i<=10;i++)
        {
          n=sc.nextInt();
          if(isPrime(n)==true)
            s+=n;
        }
        if(isPrime(s)==true)
          System.out.println("Sum of the prime numbers is also a prime number");
        else
          System.out.println("Sum of the prime numbers is not a prime number");
    }
}
```

23. Create a class called Series which will contain the following members function:

(i) long fact(int f) to find the factorial of f and return it.

(ii) void sumSeries1(int x,int n) to calculate and print the sum of the following series: x+x/2!+x/3!+x/4!+...+x/n!

(iii) void sumSeries2(int x,int n) to calculate and print the sum of the following series: $x/2!+x^2/3!+x^3/4!+x^4/5!+\ldots+x^n/(n+1)!$

(iv) void sumSeries3(int x,int n) to calculate and print the sum of the following series: $x/2! - x^2/3!+x^3/4! - x^4/5!+\ldots\pm x^n/(n+1)!$

Where the symbol ! stands for factorial eg. 5!=5*4*3*2*1, 3!=3*2*1

Ans.

```
class Series
{
  static long fact(int f)
  {
    long i,p=1;
    for(i=1;i<=f;i++)
    {
      p=p*i;
    }
    return p;
  }
  static void sumSeries1(int x,int n)
  {
    double s=0;
    int i;
    for(i=1;i<=n;i++)
    {
      s=s+(double)x/fact(i);
    }
    System.out.println("Sum="+s);
  }
```

```
    static void sumSeries2(int x,int n)
    {
       double s=0;
       int i;
       for(i=1;i<=n;i++)
       {
          s=s+Math.pow(x,i)/fact(i+1);
       }
       System.out.println("Sum="+s);
    }
    static void sumSeries3(int x,int n)
    {
       double s=0;
       int i;
       for(i=1;i<=n;i++)
       {
          if(i%2==0)
             s=s-Math.pow(x,i)/fact(i+1);
          else
             s=s+Math.pow(x,i)/fact(i+1);
       }
       System.out.println("Sum="+s);
    }
}
```

24. Using overloading technique, write methods to:

- accept two int type data as parameters and return their sum.
- accept three int type data as parameters and return their sum.
- accept two double type data as parameter and return their sum.
- accept a double type and int type as parameter and return their sum.

Ans.

```
import java.util.*;
class Sol24
{
   static int overload(int a,int b)
   {
      int s;
      s=a+b;
      return s;
   }
   static int overload(int a,int b,int c)
   {
      int s;
      s=a+b+c;
      return s;
```

```
    }
    static double overload(double a,double b)
    {
       double s;
       s=a+b;
       return s;
    }
    static double overload(double a,double b,double c)
    {
       double s;
       s=a+b+c;
       return s;
    }
}
```

25. Create a function which accepts an integer as parameter and return true if it is a prime number otherwise return false. Create another function which accepts an integer as parameter and return true if it is palindrome otherwise return false. In the main() method display all three digit pal-prime number. Palprime numbers are such numbers which are both palindrome as well as prime numbers.For example, 101,131,151,181,191,313,353,373,383,727,757,787,797, 919 and 929 are all three digit pal-prime numbers.

Ans.

```
import java.util.*;
class Sol25
{
   static boolean prime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }
   static boolean palindrome(int n)
   {
      int i,r=0,d;
      for(i=n;i>0;i=i/10)
      {
         d=i%10;
         r=r*10+d;
```

```
    }
    if(r==n)
      return true;
    else
      return false;
  }
  static void main()
  {
    int i,s;
    for(i=100;i<=999;i++)
    {
      if(prime(i)==true && palindrome(i)==true)
        System.out.println("Pal-Prime number:"+i);
    }
  }
}
```

26. Create a function which accepts an integer as parameter and return the sum of its digits. Create another function which accepts an integer as parameter and return true if it is magic number otherwise return false. In the main input an integer and check whether it is a magic number or not.

If you iterate the process of summing the decimal digits of a number and if this process terminates in 1, then the original number is called a magic number. For example 55=> (5+5)=10=>(1+0)=1.

Ans.

```
import java.util.*;
class Sol26
{
  static int sum(int a)
  {
    int i,d,s=0;
    for(i=a;i>0;i=i/10)
    {
      d=i%10;
      s=s+d;
    }
    return s;
  }

  static boolean isMagic(int n)
  {
    do
    {
      n=sum(n);
```

```
    }while(n>9);
    if(n==1)
      return true;
    else
      return false;
  }
  static void main()
  {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number:");
    n=sc.nextInt();
    if(isMagic(n)==true)
      System.out.println("Magic Number");
    else
      System.out.println("Not a Magic Number");
  }
}
```

27. Create a function which accepts an integer as parameter and return the sum of the square of its digits. Create another function which accepts an integer as parameter and return true if it is happy number otherwise return false. In the main input an integer and check whether it is a happy number or not.. For example 7=> (7^2)=49=> (4^2+9^2)=97=>(9^2 +7^2)=130 =>(1^2 +3^2+0^2)=10 =>(1^2+0^2)= 1.

Ans.

```
import java.util.*;
class Sol27
{
  static int sum(int a)
  {
    int i,d,s=0;
    for(i=a;i>0;i=i/10)
    {
      d=i%10;
      s=s+d*d;
    }
    return s;
  }

  static boolean isHappy(int n)
  {
    do
    {
      n=sum(n);
    }while(n>9);
```

```
        if(n==1)
            return true;
        else
            return false;
    }
    static void main()
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number:");
        n=sc.nextInt();
        if(isHappy(n)==true)
            System.out.println("Happy Number");
        else
            System.out.println("Not a Happy Number");
    }
}
```

28. Create a class with the following functions:

(i) boolean isBinary() which accepts a binary number as parameter and return true if it is a valid binary number or not. A binary number consists of only two digits 0 and 1.

(ii) int binToDecimal(int b) which accepts a binary number as parameter and return its decimal equivalent.

(iii) int deciToBinary(int d) which accepts a decimal number as parameter and return its binary equivalent.

(iv) void sum() where you input two binary numbers and if they are valid binary numbers find their sum.

Sample input and output when the sum() function is executed:

INPUT

Enter two valid binary numbers: 1011 11011

OUTPUT

Sum of the two given binary numbers is: 100110

Ans.

```
import java.util.*;
class Sol28
{
    static boolean isBinary(int b)
    {
        int i,d,f=0;
        for(i=b;i>0;i=i/10)
        {
            d=i%10;
            if(!(d==1 ||d==0))
                f=1;
        }
```

```
        if(f==0)
            return true;
        else
            return false;
    }

    static int binToDecimal(int b)
    {
        int i,d,s=0,c=0;
        for(i=b;i>0;i=i/10)
        {
            d=i%10;
            s=s+d*(int)Math.pow(2,c++);
        }
        return s;
    }

    static int decToBinary(int d)
    {
        int i,r,s=0,c=0;
        for(i=d;i>0;i=i/2)
        {
            r=i%2;
            s=s+r*(int)Math.pow(10,c++);
        }
        return s;
    }
    static void sum()
    {
        int a,b,s;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter 2 valid binary numbers:");
        a=sc.nextInt();
        b=sc.nextInt();
        if (isBinary(a)==false || isBinary(b)==false)
        System.out.println("Both should be valid binary numbers");
        else
        {
            s=binToDecimal(a)+binToDecimal(b);
            s=decToBinary(s);
            System.out.println("Sum of the 2 given binary number is:"+s);
        }
    }
}
```

29. Write a class with the name volume using function overloading that computes the volume of a cube, a sphere and a cuboid.

Formula:
volume of a cube (vc) = s*s*s
volume of a sphere (vs) = 4/3 * pi * r * r * r (where pi = 3.14 or 22/7)
Volume of a cuboid (vcd) = l * b * h

Ans.

```
class volume
{
   static void overload(int s)//cube
   {
      int vc;
      vc=s*s;
      System.out.println("Volume of the cube:"+vc);
   }
   static void overload(float r)//sphere
   {
      float vs;
      vs=4/3f*3.14f*r*r*r;
      System.out.println("Volume of the sphere:"+vs);
   }
   static void overload(int l,int b,int h)//cuboid
   {
      int vcd;
      vcd=l*b*h;
      System.out.println("Volume of the cuboid:"+vcd);
   }
}
```

30. Design a class to overload a function num_calc( ) as follows:

a. void num_calc(int num,char ch) with one integer argument and one character argument, computes the square of integer argument if choice ch is 's' otherwise find its cube.
b. void num_calc(int a, int b,char ch) with two integer arguments if ch is 'p' else adds the integers.
c. void num_calc(String s1,String s2) with two String arguments, which prints whether the strings are equal or not.

Ans.

```
class Overload
{
   static void num_calc(int num,char ch)
   {
      if(ch=='s')
         System.out.println("Square:"+(num*num));
      else
```

```
        System.out.println("Cube:"+(num*num*num));
    }
    static void num_calc(int a,int b,char ch)
    {
      if(ch=='p')
        System.out.println("Product:"+(a*b));
      else
        System.out.println("Sum:"+(a+b));
    }
    static void num_calc(String s1,String s2)
    {
      if(s1.equalsIgnoreCase(s2))
        System.out.println("Equal");
      else
        System.out.println("Not Equal");
    }
}
```

31. Design a class to overload a function compare() as follows:

a. void compare(int,int) - to compare two integer values and print the greater of the two integers.

b. void compare(char,char) - to compare the numeric values of two characters and print the character with higher numeric value.

c. void compare(String, String) - to compare the length of the two strings and print the longer of the two.

Ans.

```
class Sol31
{
    static void compare(int a,int b)
    {
      if(a>b)
        System.out.println(a);
      else
        System.out.println(b);
    }
    static void compare(char a,char b)
    {
      if((int)a>(int)b)
        System.out.println(a);
      else
        System.out.println(b);
    }
    static void compare(String s1,String s2)
    {
      if(s1.length()>s2.length())
```

```
            System.out.println("First string is greater");
        else
            System.out.println("second string is greater");
    }
}
```

32. Design a class to overload a function polygon() as follows:

(i) void polygon(int n, char ch): with one integer argument and one character type argument that draws a filled square of side n using the character stored in ch.

(ii) void polygon(int x, int y): with two integer arguments that draws a filled rectangle of length x and breadth y, using the symbol '@'

(iii) void polygon(): with no argument that draws a filled triangle shown below:

```
*
* *
* * *
```

Ans.

```
class Sol32
{
    static void polygon(int n,char ch)
    {
        int i,j;
        for(i=1;i<=n;i++)
        {
            for(j=1;j<=n;j++)
            {
                System.out.print(ch);
            }
            System.out.println();
        }
    }
    static void polygon(int x,int y)
    {
        int i,j;
        for(i=1;i<=y;i++)
        {
            for(j=1;j<=x;j++)
            {
                System.out.print("@");
            }
            System.out.println();
        }
    }
    static void polygon()
    {
        int i,j;
```

```
        for(i=1;i<=3;i++)
        {
          for(j=1;j<=i;j++)
          {
            System.out.print("*");
          }
          System.out.println();
        }
      }
}
```

33. Design a class to overload a function compute() as follows:

(i) void compute(int,char): to compute the square of the integer argument if the given character argument is 's' otherwise find its cube.

(ii) void compute(double char): to compute voume of a cube if the given character argument is 'v' otherwise find its diagonal.

(iii) void compute(int,int,char): to compute area of a rectangle if the given character argument is 'a' otherwise finds its perimeter.

Volume of cube=side^3 **Area of rectangle=length*breadth**

Diagonal of cube=a$\sqrt{3}$ **Perimeter of rectangle=2*(length+breadth)**

Ans.

```
class Sol33
{
  static void compute(int num,char ch)
  {
    if(ch=='s')
      System.out.println("Square:"+(num*num));
    else
      System.out.println("Cube:"+(num*num*num));
  }
  static void compute(double s,char ch)
  {
    if(ch=='v')
      System.out.println("Volume:"+(s*s*s));
    else
      System.out.println("Diagonal:"+(s*Math.sqrt(s)));
  }
  static void num_calc(int l,int b,char ch)
  {
    if(ch=='a')
      System.out.println("Area:"+(l*b));
    else
      System.out.println("Perimeter:"+(2*(l+b)));
  }
}
```

34. Design a class to overload a function series() as follows:

(i) double series(double n) with one double argument and returns the sum of the series. sum = 1/1 + 1/2 + 1/3 + ….. 1/n

(ii) double series(double a, double n) with two double arguments and returns the sum of the series. sum
= $1/a^2 + 4/a^5 + 7/a^8 + 10/a^{11}$ ….. to n terms

Ans.

```
class Sol34
{
  static double series(double n)
  {
    double s=0,i;
    for(i=1;i<=n;i++)
    {
      s+=1/i;
    }
    return s;
  }
  static double series(double a,double n)
  {
    double s=0,i,c=1;
    for(i=1;i<=n;i++)
    {
      s+=c/Math.pow(a,c+1);
      c+=3;
    }
    return s;
  }
}
```

35. Design a class to overload a function series( ) as follows:

(i) double series(double n) with one double argument and returns the sum of the series:

$$S = 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ldots \frac{1}{n!}$$

(ii) double series(double x,double n) with two double arguments and returns the sum of the series

$$S = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots \frac{x^n}{n!}$$

Ans.

```
class Sol35
{
  static double series(double n)
```

```
    {
      double s=0,i,j,f;
      for(i=1;i<=n;i++)
      {
        f=1;
        for(j=1;i<=i;j++)
        {
          f=f*j;
        }
        s+=1/f;
      }
      return s;
    }
    static double series(double x,double n)
    {
      double s=0,i,j,f;
      for(i=1;i<=n;i++)
      {
        f=1;
        for(j=1;i<=i;j++)
        {
          f=f*j;
        }
        s+=Math.pow(x,i)/f;
      }
      return s;
    }
}
```

36. Write a menu driven program to accept a number from the user and check whether is a 'BUZZ' number or to accept any two numbers and print 'GCD' of them.

a. A BUZZ number is the number which either ends with 7 or divisible by 7.

b. GCD (Greatest Common Divisor) of two integers is calculated by continued division method. Divide the larger number by the smaller, the remainder then divides the previous devisor. The process is repeated till the remainder is zero. The divisor then results the GCD.

Ans.

```
import java.util.*;
class Sol36
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int i,n,r,ch,d;
```

```
        System.out.println("M E N U");
        System.out.println("1. Buzz Number");
        System.out.println("2. GCD");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
           case 1:
              System.out.println("Enter a number:");
              n=sc.nextInt();
              if(n%10==7 || n%7==0)
                 System.out.println("Buzz Number");
              else
                 System.out.println("Not a Buxx number");
              break;
           case 2:
              System.out.println("Enter 2 numbers:");
              n=sc.nextInt();
              d=sc.nextInt();
              do
              {
                 r=n%d;
                 if(r!=0)
                 {
                    n=d;
                    d=r;
                 }
              }while(r!=0);
              System.out.println("GCD:"+d);
              break;
           default:
              System.out.println("Invalid choice!");
        }
     }
  }
```

37. Write a menu driven program to accept a number and check and display whether it is a Prime Number or not OR an Automorphic Number or not. (Use switch-case statement).

a. Prime number: A number is said to be a prime number if it is divisible only by 1 and itself and not by any other number. Example : 3,5,7,11,13, etc.

b. Automorphic number: An automorphic number is the number which is contained in the last digit(s) of its square. Example: 25 is an automorphic number as its square is 625 and 25 is present as the last two digits.

Ans.

```
import java.util.*;
class Sol37
{
   static boolean isPrime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }
   static boolean isAutomorphic(int n)
   {
      int i,c=0;
      for(i=n;i>0;i=i/10)
      {
         c++;
      }
      if(n==(n*n)%(int)Math.pow(10,c))
         return true;
      else
         return false;
   }
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,n,r,ch,d;
      System.out.println("M E N U");
      System.out.println("1. Prime Number");
      System.out.println("2. Automorphic");
      System.out.println("Enter your choice:");
      ch=sc.nextInt();
      switch(ch)
   {
        case 1:
```

```
            System.out.println("Enter a number:");
            n=sc.nextInt();
            if(isPrime(n)==true)
            System.out.println("Prime Number");
            else
            System.out.println("Not a prime number");
            break;
        case 2:
            System.out.println("Enter a number:");
            n=sc.nextInt();
            if(isAutomorphic(n)==true)
            System.out.println("Automorphic Number");
        else
                System.out.println("Not an Automorphic number");
        break;
        default:
            System.out.println("Invalid choice!");
      }
   }
}
```

38. Write a menu driven program to perform the following: (Use switch case statement)

a. To print the series 0,3,8,15,24… n terms(value of 'n' is to be an input by the user).

b. To find the sum of the series given below:
S= 1/2 + 3/4 + 5/6 + 7/8 …. 19/20.

Ans.

```
import java.util.*;
class Sol38
{
   static void series(int n)
   {
      int i,s=0,c=3;
      for(i=1;i<=n;i++)
      {
         System.out.print(s+" ");
         s=s+c;
         c=c+2;
      }
   }
   static void sumSeries()
   {
```

```
        int i;
        float s=0;
        for(i=1;i<=19;i+=2)
        {
          s+=(float)i/(i+1);
        }
        System.out.print("Sum:"+s);
    }
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int ch,n;
        System.out.println("M E N U");
        System.out.println("1. Display series");
        System.out.println("2. sum of the series");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
          case 1:
            System.out.println("Enter a number:");
            n=sc.nextInt();
            series(n);
            break;
         case 2:
            sumSeries();
            break;
         default:
            System.out.println("Invalid choice!");
        }
    }
}
```

39. Using the switch statement, write a menu-driven program to:

(i) Generate and display the first 10 terms of the Fibonacci series 0,1,1,2,3,5,….

(ii) find the sum of the digits of an integer that is input. Eg 15390=18

For an incorrect choice, appropriate error message should be displayed.

Ans.

```
import java.util.*;
class Sol39
{
```

```
static void fibo()
{
   int i,f=1,s=0,t=0;
   System.out.print(t+" ");
   for(i=1;i<=9;i++)
   {
      t=f+s;
      System.out.print(t+" ");
      f=s;
      s=t;
   }
}
static void sum(int n)
{
   int i,d,s=0;
   for(i=n;i>0;i/=10)
   {
      d=i%10;
      s+=d;
   }
   System.out.print("Sum:"+s);
}
static void main()
{
   Scanner sc=new Scanner(System.in);
   int ch,n;
   System.out.println("M E N U");
   System.out.println("1. Fibonacci series");
   System.out.println("2. Sum of the digits in a number");
   System.out.println("Enter your choice:");
   ch=sc.nextInt();
   switch(ch)
   {
      case 1:
         fibo();
         break;
     case 2:
         System.out.println("Enter a number:");
         n=sc.nextInt();
         sum(n);
         break;
```

```
            default:
                System.out.println("Invalid choice!");
        }
    }
}
```

40. Using the switch statement, write a menu driven program:

(i) To check and display whether a number input by the user is a composite number or not (A number is said to be a composite, if it has one or more then one factors excluding 1 and the number itself).
Example: 4, 6, 8, 9…

(ii) To find the smallest digit of an integer that is input:
Sample input: 6524
Sample output: Smallest digit is 2
For an incorrect choice, an appropriate error message should be displayed.

Ans.

```
import java.util.*;
class Sol40
{
  static void isComposite(int n)
  {
    int i,c=0;
    for(i=1;i<=n;i++)
    {
      if(n%i==0)
        c++;
    }
    if(c>2)
      System.out.println("Composite number");
    else
      System.out.println("Not a Composite number");
  }
  static void smallest(int n)
  {
    int i,d,min=0;
    for(i=n;i>0;i=i/10)
    {
      d=i%10;
      if(min==0)
        min=d;
      if(d<min)
        min=d;
    }
    System.out.println("Smallest digit:"+min);
```

```
    }
    static void main()
    {
       Scanner sc=new Scanner(System.in);
       int i,n,r,ch,d;
       System.out.println("M E N U");
       System.out.println("1. Composite Number");
       System.out.println("2. Smallest digit in a number");
       System.out.println("Enter your choice:");
       ch=sc.nextInt();
       switch(ch)
       {
          case 1:
             System.out.println("Enter a number:");
             n=sc.nextInt();
             isComposite(n);
             break;
         case 2:
             System.out.println("Enter a number:");
             n=sc.nextInt();
             smallest(n);
             break;
         default:
             System.out.println("Invalid choice!");
       }
    }
}
```

Chapter 5
Constructors

Mental Drill

Tick (✓) the correct option.

1. Which of the following is not applicable for a constructor function?

   a. It has the same name as the class.

   b. It has no return-type

   c. It is usually used for initialisation.

   d. It can be invoked using an object like any other member function.

   Ans. d. It can be invoked using an object like any other member function.

2. If the name of a class is 'Number', what can be the possible name for its constructor?

   a. Number
   b. number
   c. No
   d. No

   Ans. a. Number

3. Which among the following is a type of constructor?

   a. Parameterised constructor
   b. Non-parameterised constructor
   c. Both a and b
   d. None of these

   Ans. c. Both a and b

4. If constructors are overloaded, what differentiates it?

   a. Parameter list
   b. Return type
   c. Both a and b
   d. None of these

5. What access specifier for a constructor allows you to create an object only within the class?

   a. public
   b. private
   c. protected
   d. default

   Ans. b. private

6. Name the type of constructor that gets invoked when an object is created, which is initialised with the content of another object.

   a. Copy constructor
   b. Default constructor
   c. Overloaded constructor
   d. None of these

   Ans. a. Copy constructor

7. Categorise the type of object that can be created without any name or identifier.

a. Temporary object
b. Anonymous object
c. Both a and b
d. None of these

Ans. c. Both a and b

8. Predict the output of the following program:

```
class T
    {
            int t = 20;
            T()
        {
            t = 40;
        }
            public static void main(String args[])
        {
            T t1 = new T();
            System.out.println(t1.t);
        }
    }
```

a. 20
b. 40
c. Compiler Error
d. None of these

Ans. b. 40

9. The following code contains one compilation error, find it?

```
public class Test {
Test() { } // line 1
static void Test() { this(); } // line 2
public static void main(String[] args) { // line 3
Test(); // line 4
}
}
```

a. At line 1, constructor Tester must be marked public like its class

b. At line 2, constructor call

c. At line 3, compilation error, ambiguity problem, compiler can't determine whether a constructor

d. At line 4

Ans. b. At line 2, constructor call

10. Which of the following is not true for static block?

a. It is used to initialise static variables.

b. It gets executed when a class gets loaded in the memory.

c. It can print the content of instance variables.

d. It begins with the static keyword.

Ans. c. It can print the content of instance variables.

SECTION A

1. What is a constructor? Why is it needed in a program?

Ans. A constructor in Java is a block of code similar to a method that is called when an instance of an object is created. A constructor is needed to initialise data members with legal initial values.

2. State the characteristics of a constructor.

Ans. The characteristics of a constructor are:

a. It has the same name as the class-name.
b. It does not have any return type.
c. It follows the usual rules of accessibility as other members of a class and therefore access modifiers can be applied to it.
d. It gets called automatically, whenever an object is created.

3. How are constructors invoked?

Ans. Constructor function gets called (invoked) automatically whenever an object is created.

4. Why do we need a constructor as a class member?

Ans. A constructor is a special member method which will be called by the JVM implicitly for placing user/programmer defined values instead of placing default values. Constructors are meant for initializing the object.

5. State the difference between function and constructor.

Ans. Following are the difference between constructor and method.

a. Constructor is used to initialize an object whereas method is used to exhibits functionality of an object.
b. Constructors are invoked implicitly whereas methods are invoked explicitly.
c. Constructor does not return any value where the method may/may not return a value.
d. In case constructor is not present, a default constructor is provided by java compiler. In the case of a method, no default method is provided.
e. Constructor should be of the same name as that of class. Method name should not be of the same name as that of class.

6. How are private constructors called?

Ans. Private constructors allows an object to be created only within the methods of the class where it is private.

7. What are the different types of constructors available in Java?

Ans. Parameterised constructor and Non-parameterised constructor.

8. What is a default constructor?

Ans. When we do not explicitly define a constructor for a class, then java creates a default constructor for the class. It is essentially a non-parameterized constructor, i.e. it doesn't accept any arguments. The default constructor's job is to call the super class constructor and initialize all instance variables.

9. Point out the difference between implicit and explicit default constructors.

Ans. Implicit default constructor is the default constructor created by java if the user do not create a constructor. Explicit default constructor is a non-parameterised constructor defined by the user to initialise data members with legal initial values.

10. What are temporary objects? How are they created, explain with the help of an example?

Ans. Temporary or anonymous objects or instances are the ones that live in the memory as long as it is being used or referenced in an expression and after that it dies. Temporary objects are created by explicit call to a constructor, preceded with the new command. For example,

```
public class temp
{
        int a,b;
        temp(int x, int y)
        {
            a=x;
            b=y;
        }
        void show()
        {
            System.out.println(a+","+b);
        }
            static void test( )
        {
            new temp(1,2).show( );
        }
}
```

Here the statement new temp(1,2) of the statement new temp(1,2).show(); creates an anonymous temporary object and lives in the memory as long as show() of the statement new temp(1,2).show(); is being executed. Upon completion of the execution of the show() function, the temporary object gets wiped out of the memory.

11. What is constructor overloading? Illustrate it with the help of an example.

Ans. Just like method overloading, constructors also can be overloaded. Same constructor declared with different parameters in the same class is known as constructor overloading. Compiler differentiates which constructor is to be called depending upon the number of parameters and their sequence of data types.

```
Java
public class Perimeter
```

```
{
 public Perimeter()                                    // I
 {
  System.out.println("From default");
 }
 public Perimeter(int x)                               // II
 {
  System.out.println("Circle perimeter: " + 2*Math.PI*x);
 }
 public Perimeter(int x, int y)                        // III
 {
  System.out.println("Rectangle perimeter: " +2*(x+y));
 }
 public static void main(String args[])
 {
  Perimeter p1 = new Perimeter();                  // I
  Perimeter p2 = new Perimeter(10);                // II
  Perimeter p3 = new Perimeter(10, 20);            // III
 }
}
```

12. What is a destructor? Is destructor function necessary in Java? If no, explain why?

Ans. A destructor is a special method called automatically during the destruction of an object. In java a destructor is not necessary because Java is a garbage collected language you cannot predict when (or even if) an object will be destroyed. Hence there is no direct equivalent of a destructor.

13. What is the implicit return type of a constructor function?

Ans. The implicit return type is the class itself.

14. Enter any two variables through constructor with parameters and write a program to swap and print the values.

Ans.
```
class Swap
{
    int a,b;
    Swap(int x,int y)
    {
        a=x;
        b=y;
    }
    void interchange()
    {
        int t=a;
        a=b;
        b=t;
    }
}
```

15. What is the default initial value of a boolean variable data type?

Ans. the default initial value is 'false'.

B. Consider the following code and answer the questions that follow:

```
class Myclass
{
        int a,b;
        void Myclass(int x, int y)
        {
            a=x;
            b=y;
        }
        int Myclass(int x)
        {
            a=b=x;
        }
        void show( )
        {
            System.out.println(a+ " "+y);
        }
        public static void main(String args[])
        {
            Myclass ob1=new Myclass();
            Myclass ob2=new Myclass(12.3,14.6,15);
            Myclass ob3=new Myclass(7);
            ob1.ob2.ob3.show();
        }
}
```

Ans. The program has errors and here is the solution:

```
class Myclass
{
        int a,b;
        void Myclass(int x, int y)
        {
            a=x;
            b=y;
        }
        int Myclass(int x)
        {
            a=b=x;
        }
        void show( )
        {
```

```
            System.out.println(a+ " "+b);
        }
        public static void main(String args[])
        {
            Myclass ob2=new Myclass(12,15);
            Myclass ob3=new Myclass(7);
            ob2.show();
            ob3.show();
        }
}
```

C. Consider the following code and answer the questions that follow:

```
class academic
{
    int x,y;
    void access()
    {
        int a,b;
        academic student=new academic();
        System.out.println(Object Created");
    }
}
```

a. What is the object name of the class?

b. Name the instance variables used in the class.

c. Write the name of local variables used in the program.

d. Give the type of function being used and its name.

Ans.

a. student
b. x and y
c. a and b
d. procedural function Name: access

D. State the output of the following program segment:

```
class Today
{
        static int a;
        char b;
        void input()
        {
            a=20;
            b='Z';
        }
        void convert()
        {
            char c=(char)(a+b);
```

```
            System.out.println(c);
        }
        public static void main()
        {
            Today t=new Today();
            t.input();
            t.convert();
        }
}
```

Based on the above given piece of code, answer that questions the follow:

i. Name the instance, class and local variables.

ii. What is the name of the constructor of the above class?

iii. Explain the line: Today t=new Today();

Ans.

Output:

n

i. Instance variable: b
   Class variable: a
   Local variable: c

ii. Today()

iii. An object named t is being created of 'Today' type.

E. In the program given below, state the name and value of the

i. method argument or argument variable

ii. class variable

iii. local variable

iv. instance variable

```
class Myclass
{
        static int x=7;
        int y=2;
        public static void main(String args[])
        {
            Myclass obj=new Myclass();
            System.out.println(x);
            obj.sampleMethod(5);
            int a=6;
            System.out.pritnln(a);
        }
            void sampleMethod(int n)
            {
                System.out.println(n);
                System.out.println(y);
            }
}
```

Ans.

i. main() method: String args[]
   sampleMethod() method: int n
ii. x
iii. a
iv. y

SECTION B

Write programs for the following:

1. Define a class called Box, having members as:

Data Members: length, breadth and height of int type.
Member Functions:
i. Constructor to initialise the data members.
ii. To compute and display the volume.

Ans.

```
class Box
{
   int length,breadth,height;
   Box(int l,int b,int h)
   {
      length=l;
      breadth=b;
      height=h;
   }
   void compute()
   {
      int vol;
      vol=length*breadth*height;
      System.out.println("Volume:"+vol);
   }
}
```

2. Define a class called Friend, having members as:

Data Members: Name, Address, Favourite hobby
Member Functions
i. Constructor to initialise the data members.
ii. To display the details.
Also create the main() and create 2 objects in it and initialise it with information of two of your friends and display it, by calling the above functions.

Ans.

```
class Friend
{
   String Name,Address,hobby;
   Friend(String n,String a,String h)
   {
      Name=n;
      Address=a;
      hobby=h;
   }
   void display()
   {
      System.out.println("Name:"+Name);
      System.out.println("Address:"+Address);
      System.out.println("Favourite hobby:"+hobby);
   }
   public static void main(String args[])
   {
      Friend ob1=new Friend("Soumen","Delhi","Philately");
      Friend ob2=new Friend("Sujit","Patna","Guitar");
      ob1.display();
      ob2.display();
   }
}
```

3. Define a class named four Side, having members as:

Data members: length, breadth

Member Functions:

i. Overloaded constructor to initialise the dimension of the four-sided figure with a square and a rectangle.

ii. Compute the area and display it.

Also create the main() method to show the implementation of the above methods.

Ans.

```
class FourSide
{
   int length,breadth;
   FourSide(int s)
   {
      length=breadth=s;
   }
   FourSide(int l,int b)
   {
      length=l;
      breadth=b;
```

```
    }
    void compute()
    {
       int area;
       area=length*breadth;
       System.out.println("Area:"+area);
    }
    public static void main(String args[])
    {
       FourSide ob1=new FourSide(12);
       FourSide ob2=new FourSide(17,6);
       ob1.compute();
       ob2.compute();
    }
}
```

4. The sum of n natural numbers is defined by the following formula:

$$\frac{n*(n+1)}{1}$$

Create a class named Natural, which will contain the following members:
Data members: n and s of int data type.
Member functions:

i. Parametrised constructor to initialise n.
ii. void compute( ) to calculate the sum of first n natural numbers using the above formula and store it in s.
iii. void display( ) to show the sum.

Ans.

```
class Natural
{
   int n,s;
   Natural(int t)
   {
      n=t;
   }
      void compute()
   {
      int i;
      s=0;
      for(i=1;i<=n;i++)
         s+=i;
   }
   void display()
   {
```

```
        System.out.println("Sum:"+s);
    }
}
```

5. Define a class named Conversion having the following static methods:

i. int mTocm(int m), which converts metre(m) to centimeter and return it.

ii. int multiply(int x, int y), which returns the product of x and y.

Define another class named Rectangle which contains the following data members:

- **length of int type which holds the length of a rectangle in metre.**
- **breadth of int type which holds the breadth of a rectangle in centimeter.**

Create member functions:

i. Constructor to initialise the data members.

ii. Convert the length to centimetre by calling the relevant method of the above class.

iii. Compute the area by calling the relevant method of the above class and display the result.

Ans.

```
class Conversion
{
    int n,s;
    int mTocm(int m)
    {
        return m*100;
    }
    int multiply(int x,int y)
    {
        return x*y;
    }
}
public class Rectangle
{
    int length,breadth;
    Rectangle(int m,int cm)
    {
        length=m;
        breadth=cm;
    }
    void convert()
    {
        length=new Conversion().mTocm(length);
    }
    void compute()
    {
        int area;
        convert();
```

```
        area=new Conversion().multiply(length,breadth);
        System.out.println("Area:"+area);
    }
}
```

6. A class Palin has been defined to check whether a positive number is a Palindrome number or not.

The number 'N' is palindrome if the original number and its reverse are same. Some of the members of the class are given below:

Classname :

Palin Data members/instance variables:

- **num : integer to store the number**
- **revnum : integer to store the reverse of the number**

Methods/Member functions:

- **Palin( ) : constructor to initialize data members with legal initial values.**
- **void accept( ) : to accept the number.**
- **int reverse(int y) : reverses the parameterized argument 'y' and stores it in 'revnum'.**
- **void check( ) : checks whether the number is a Palindrome by invoking the function reverse( ) and display the result with an appropriate message.**

Specify the class Palin giving the details of the constructor( ), void accept( ),int reverse(int) and void check( ). Define the main( ) function to create an object and call the functions accordingly to enable the task.

Ans.

```
import java.util.*;
class Palin
{
    int num,revnum;
    Palin()
    {
        num=0;
        revnum=0;
    }
    void accept()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number:");
        num=sc.nextInt();
    }
    int reverse(int y)
    {
        for(int i=num;i>0;i/=10)
        {
            int d=i%10;
```

```
            revnum=revnum*10+d;
        }
        return revnum;
    }
    void check()
    {
        if(num==reverse(num))
            System.out.println("Palindrome");
        else
            System.out.println("Not palindrome");
    }
    public static void main(String args[])
    {
        Palin ob=new Palin();
        ob.accept();
        ob.check();
    }
}
```

7. Design a class named Numbers, which will contain the following members:

Data Members: a, b and c of int data type.

Member Functions:

i. Parameterised constructor to initialise a and b.

ii. void show( ) to display the contents of a, b and c.

iii. void compute( ) to add a and b and store it in c.

In the main( ) create an object and initialise a and b with any value and add a and b by invoking the compute( ) function and display the contents of a, b and c using show( ) function.

Ans.

```
class Numbers
{
    int a,b,c;
    Numbers(int x,int y)
    {
        a=x;
        b=y;
    }
    void show()
    {
        System.out.println(a+" "+b+" "+c);
    }
    void compute()
    {
        c=a+b;
```

```
    }

    public static void main(String args[])
    {
      Numbers ob=new Numbers(12,15);
      ob.compute();
      ob.show();
    }
}
```

8. Create a class named Number with the following members:

Data members: a of int data type.

Member Functions:

- **Parameterised constructor to initialise the data member.**
- **Default constructor to initialise the data member with 0.**
- **To display a only.**
- **Which accepts an instance of Number class as parameter and creates another object, whose a will contain the sum the a's of the current and the passed object. This function should return the object newly instantiated.**

Also create a main( ) with at least 3 objects to show its working.

Ans.

```
class Number
{
  int a;
  Number(int x)
  {
    a=x;
  }
  Number()
  {
    a=0;
  }
  void show()
  {
    System.out.println(a);
  }
  Number sum(Number t)
  {
    Number x=new Number();
    x.a=a+t.a;
    return x;
  }
```

```
    public static void main(String args[])
    {
      Number ob1=new Number(12);
      Number ob2=new Number(15);
      Number ob3;
      ob3=ob1.sum(ob2);
      ob3.show();
    }
}
```

9. You are to print the telephone bill of a subscriber. Create a class having the following data members:

Phone Number of long data type (for storing the phone number).

Name of String type (for storing the name of a the subscriber).

Hire Charge a symbolic constant of int type (to store monthly hire charge say ₹. 200).

Units Consumed of int type (to store the monthly units consumed by the subscriber).

Bill Amount of float type (to store the bill amount that is payable).

Create member functions for the following:

i. Constructor to initialise all data members except Hire Charge and Bill Amount.

ii. Calculate the bill amount payable which is Hire Charge+(₹. 1 per unit for the first 100 units, ₹. 1.50 per unit for the next 100 units and ₹. 2.00 per unit there after.

iii. To display the Bill for the subscriber.

Ans.

```
import java.util.*;
class Telephone
{
  long pno;
  String name;
  int hire;
  int units;
  float bill;
  Telephone(long p,String n,int u)
  {
    pno=p;
    name=n;
    units=u;
  }
  void calculate()
  {
    hire=200;
    bill=0;
    if(units<=100)
```

```
        bill=hire+units*1;
      else if(units>100 && units<=200)
        bill=hire+100*1+(units-100)*1.50f;
      else
        bill=hire+100*1+100*1.50f+(units-200)*2;
    }
    void show()
    {
      System.out.println("BILL");
      System.out.println("Phone No.:"+pno);
      System.out.println("Name:"+name);
      System.out.println("Hire Charge:"+hire);
      System.out.println("Units consumed:"+units);
      System.out.println("Bill:"+bill);
    }
}
```

10. Define a class Taximeter having the following description:

Data members/instance variables:

int taxino	-	to store taxi number
String name	-	to store passenger's name
int km	-	to store number of kilometers travelled

Member functions:

Taximeter()	-	constructor to initialise taxino to 0, name to "" and km to 0.
input()	-	to store taxino,name,km
calculate()	-	to calculate bill for a customer according to given conditions

Kilometers travelled(km)	**Rate/km**
1 km	₹ 25
1 < km ≤ 6	₹ 10
6 < km ≤ 12	₹ 15
12 < km ≤ 18	₹ 20
>18 km	₹ 25

display()- To display the details in the following format

Taxino Name Kilometres travelled Bill amount

Ans.

```
import java.util.*;
class Taximeter
{
  int taxino;
  String name;
  int km;
  Taximeter()
  {
    taxino=0;
```

```
    name=" ";
    km=0;
  }
  void input()
  {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the taxi no.:");
    taxino=sc.nextInt();
    System.out.println("Enter the name:");
    name=sc.nextLine();
    System.out.println("Enter the kilometer:");
    km=sc.nextInt();
  }
  int calculate()
  {
    int bill=0;
    if(km==1)
      bill=25*km;
    else if(km>1 && km<=6)
      bill=10*km;
    else if(km>6 && km<=12)
      bill=15*km;
    else if(km>12 && km<=18)
      bill=20*km;
    else
      bill=25*km;
    return bill;
  }
  void display()
  {
    System.out.println("Taxino\t\tName\t\tKilomtres travelled\t\tBill amount");
    System.out.println(taxino+"\t\t"+name+"\t\t"+km+"\t\t"+calculate());
  }
}
```

11. Create a class which will contain the following components:

Data Members: a and b of int type.

Member Functions:

i. Constructor to initialise a and b.

ii. Return the sum of a and b.

iii. To display the value of a and b.

Also create a static function which accepts two objects as parameters and print a and b of the object whose sum is the maximum.

In the main() create two objects, initialise them and display the data members of that object whose sum of the data-members is the maximum.

Ans.

```
import java.util.*;
class Numbers
{
   int a,b;
   Numbers(int x,int y)
   {
      a=x;
      b=y;
   }
   int sum()
   {
      return a+b;
   }
   void display()
   {
      System.out.println("a="+a+"b="+b);
   }
   static void compare(Numbers ob1,Numbers ob2)
   {
      if(ob1.sum()>ob2.sum())
         ob1.display();
      else
         ob2.display();
   }
   public static void main(String args[])
   {
      Scanner sc=new Scanner(System.in);
      int x,y;
      System.out.println("Enter 2 numbers:");
      x=sc.nextInt();
      y=sc.nextInt();
      Numbers ob1=new Numbers(x,y);
      System.out.println("Enter 2 numbers:");
      x=sc.nextInt();
      y=sc.nextInt();
      Numbers ob2=new Numbers(x,y);
      compare(ob1,ob2);
   }
}
```

12. Define a class called FruitJuice with the following description:

Instance variables/data members:

int product_code - stores the product code number

String flavour - stores the flavour of the juice.(orange, apple, etc.)

String pack_type - stores the type of packaging (tetra-pack, bottle, etc.)
int pack_size - stores package size (200ml, 400ml, etc.)
int product_price - stores the price of the product

Member Methods:

FriuitJuice() - default constructor to initialise integer data members to zero and string data members to "".
void input() - to input and store the product code, flavour, pack type, pack size and product price.
void discount() - to reduce the product price by 10.
void display() - to display the product code, flavour, pack type, pack size and product price.

Create an object in the main method and call all the above methods in it.

Ans.

```
import java.util.*;
class FruitJuice
{
   int product_code,pack_size,product_price;
   String flavour,pack_type;
    FruitJuice()
    {
      product_code=pack_size=product_price=0;
      flavour=pack_type="";
    }
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the product code:");
      product_code=sc.nextInt();
      System.out.println("Enter the flavour:");
      flavour=sc.nextLine();
      System.out.println("Enter the pack type:");
      pack_type=sc.nextLine();
      System.out.println("Enter the pack size:");
      pack_size=sc.nextInt();
      System.out.println("Enter the product price:");
      product_price=sc.nextInt();
   }
   void discount()
   {
      product_price-=10;
   }
   void display()
   {
      System.out.println("Product code:"+product_code);
```

```
        System.out.println("Flavour:"+flavour);
        System.out.println("Pack type:"+pack_type);
        System.out.println("Pack size:"+pack_size);
        System.out.println("Product price:"+product_price);
    }
    public static void main(String args[])
    {
        FruitJuice ob=new FruitJuice();
        ob.input();
        ob.discount();
        ob.display();
    }
}
```

13. Create a class SalaryCalculation that is described as below:

Class Name : SalaryCalculation

Data members : name (String type data)

basicPay, specialAlw, conveyanceAlw, gross, pf, netSalary AnnualSal (All double type data)

Member methods :

i. SalaryCalculation( ) - A constructor to assign name of employee (name), basic salary (basicPay) of your choice and conveyance allowance (conveyanceAlw) As ₹ 1000.00

ii. void SalaryCal( ) - to calculate other allowances and salaries as given:
specialAlw = 25% of basic salary.
gross = basicPay + specialAlw + conveyanceAlw.
netSalary = gross - pf.
AnnualSal = 12 months netSalary.

iii. void display( ) - to print the name and other calculations with suitable headings.

Write a program in Java to calculate all the details mentioned above and print them all.

Ans.

```
import java.util.*;
class SalaryCalculation
{
    String name;
    double basicPay, specialAlw, conveyanceAlw, gross, pf, netSalary, AnnualSal;
    SalaryCalculation()
    {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the name:");
      name=sc.nextLine();
      System.out.println("Enter the basic pay:");
      basicPay=sc.nextDouble();
      conveyanceAlw=1000.00;
    }
   void SalaryCal()
```

```
    {
      specialAlw=25/100.0*basicPay;
      pf=10/100.0*basicPay;
      gross = basicPay + specialAlw + conveyanceAlw;
      netSalary=gross-pf;
      AnnualSal=12*netSalary;
    }
    void display()
    {
      System.out.println("Name:"+name);
      System.out.println("Basic Pay:"+basicPay);
      System.out.println("Special Allwance:"+specialAlw);
      System.out.println("Conveyance allowance :"+conveyanceAlw);
      System.out.println("Gross:"+gross);
      System.out.println("Provident fund:"+pf);
      System.out.println("Net Salary:"+netSalary);
      System.out.println("Annual Salary:"+AnnualSal);
    }
    public static void main(String args[])
    {
      SalaryCalculation ob=new SalaryCalculation();
      ob.SalaryCal();
      ob.display();
    }
}
```

14. A class Compound is created to calculate the compound interest using:

$CI = P = \left(1+\frac{r}{100}\right)^{1} - P$ **where P - is the Principal amount, r - rate of interest and t-time period in years.**

Data memebers of class: pamt, rate (double data type to store principal amount and rate of interest), time (integer to store time period)

Functions of the class:

i. Compound()-constructor to assign default values to all the data memebers.

ii. void input()-to input the principal, rate and time from the user.

iii. double findInterest()-to find and return compound interest using the given formula.

iv. void printData()-to print the principal, rate and time.

Write a main function to input required data and by invoking suitable functions print the entered data and compound interest.

Ans.

```
import java.util.*;
class Compound
{
```

```
    double pamt, rate;
    int time;
    Compound()
    {
      pamt=rate=0.0;
      time=0;
    }
    void input()
    {
     Scanner sc=new Scanner(System.in);
     System.out.println("Enter the principal:");
     pamt=sc.nextDouble();
     System.out.println("Enter the rate:");
     rate=sc.nextDouble();
     System.out.println("Enter the time:");
     time=sc.nextInt();
    }
   double findInterest()
   {
      double ci;
      ci=pamt*Math.pow(1+rate/100.0,time)-pamt;
      return ci;
   }
   void printData()
   {
      System.out.println("Principal:"+pamt);
      System.out.println("Rate:"+rate);
      System.out.println("Time:"+time);
   }
   public static void main(String args[])
   {
      Compound ob=new Compound();
      ob.input();
      ob.printData();
      System.out.println("Compound Interest:"+ob.findInterest());
   }
}
```

15. Define a class PhoneBill with the following descriptions.

Data members:

customerName	of type character array
phoneNumber	of type long
no_of_units	of type int
rent	of type int
amount	of type float.

Member Functions:

i. calculate( ) This member function should calculate the value of amount as rent+ cost for the units.

where cost for the units can be calculated according to the following conditions.

No_of_units	Cost
First 50 calls	Free
Next 100 calls	0.80 @ unit
Next 200 calls	1.00 @ unit
Remaining calls	1.20 @ unit

ii. A constructor to assign initial values of customerName as "Raju", phoneNumber as 259461, no_of_units as 50, rent as 100, amount as 100.

iii. A function accept( ) which allows user to enter customerName, phoneNumber, no_of_units and rent and should call function calculate( ).

iv. A function Display( ) to display the values of all the data members on the screen.

Ans.

```
import java.util.*;
class PhoneBill
{
   String customerName;
   long phoneNumber;
   int no_of_units,rent;
   float amount;
   void calculate()
   {
      float cost=0;
      if(no_of_units<=50)
        cost=0;
      else if(no_of_units>50 && no_of_units<=150)
        cost=50*0+(no_of_units-50)*0.80f;
      else if(no_of_units>150 && no_of_units<=350)
        cost=50*0+100*0.80f+(no_of_units-150)*1.00f;
   }
   PhoneBill()
   {
      customerName="Raju";
      phoneNumber=259461;
      no_of_units=50;
      amount=rent=100;
   }
   void accept()
   {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the customer name:");
    customerName=sc.nextLine();
```

```
        System.out.println("Enter the phone number:");
        phoneNumber=sc.nextLong();
        System.out.println("Enter the no. of units:");
        no_of_units=sc.nextInt();
        System.out.println("Enter the rent:");
        rent=sc.nextInt();
      }

    void Display()
    {
        System.out.println("Customer name:"+customerName);
        System.out.println("Phone number:"+phoneNumber);
        System.out.println("No. of units:"+no_of_units);
        System.out.println("Rent:"+rent);
        System.out.println("Amount:"+amount);
    }
    public static void main(String args[])
    {
        PhoneBill ob=new PhoneBill();
        ob.accept();
        ob.calculate();
        ob.Display();
    }
}
```

16. Define a class Sports with the following descriptions:

Data members:
s_code of type long
s_name of type (String)
fees of type integer
duration of type integer
Member Functions:

i Constructor to assign initial values of s_code as 1001, s_name as "Cricket", fees as 500, duration as 70.

ii. A function newSports() which allows user to enter s_code, s_name and duration. Also assign the values to fees as per the following conditions:

s_name	**Fees**
Table Tennis	2000
Swimming	4000
Football	3000

iii. A function displaySports() to display all the details.

Ans.

```
import java.util.*;
class Sports
{
```

```
    String s_name;
    long s_code;
    int fees,duration;
    Sports()
    {
       s_name="Cricket";
       s_code=1001;
       fees=500;
       duration=70;
    }
    void newSports()
    {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the code:");
      s_code=sc.nextLong();
      System.out.println("Enter the sports name:");
      s_name=sc.nextLine();
      System.out.println("Enter the duration:");
      duration=sc.nextInt();
      if(s_name.equalsIgnoreCase("Table Tennis"))
       fees=2000;
      else if(s_name.equalsIgnoreCase("Swimming"))
       fees=4000;
      if(s_name.equalsIgnoreCase("Football"))
       fees=3000;
    }
   void displaySports()
   {
      System.out.println("Code:"+s_code);
      System.out.println("Sports name:"+s_name);
      System.out.println("Duration:"+duration);
      System.out.println("Fees:"+fees);
   }
}
```

17. Create a class called GeneratePrime which will be used to generate n number of prime numbers. The class should have the following members:

Data Members:
n of int data type.
Member Functions:

i. Parameterised constructor to initialise the value of n.
ii. Method called isPrime() which accepts an integer as a parameter and return true if it is a prime number otherwise return false.
Method called display() which displays the first n prime number by calling the above function.

Ans.

```
import java.util.*;
class GeneratePrime
{
   int n;
   GeneratePrime(int a)
   {
      n=a;
   }
   boolean isPrime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }

   void display()
   {
      int i=0,p=2;
      while(i<n)
      {
         if(isPrime(p)==true)
         {
           System.out.println(p);
           i++;
         }
         p++;
      }
   }
}
```

18. Create a class called PrimeDigits to find the sum of the prime digits in an integer. The class should have the following members.

Data Members:
n of int data type.
Member Functions:
i. Parameterised constructor to initialise the value of n.

ii. Method called isPrime() which accepts an integer as a parameter and return true if it is a prime number otherwise return false.
Method called sumOfPrimeDigits() which accepts an integer as a parameter and find the sum of prime digits only.

Ans.

```
import java.util.*;
class PrimeDigits
{
   int n;
   PrimeDigits(int a)
   {
      n=a;
   }
   boolean isPrime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
      }
      if(c==2)
         return true;
      else
         return false;
   }

   void sumOfPrimeDigits(int n)
   {
      int i=n,d,s=0;
      while(i>0)
      {
         d=i%10;
         if(isPrime(d)==true)
            s=s+d;
         i/=10;
      }
      System.out.println("Sum of prime digits:"+s);
   }
}
```

19. Create a class called Series which will contain the following members:

Data Members:
x of double type
n of int type

Member Functions:

i. Parameterised constructor to initialise the data members.

ii. To calculate and print the sum of the following series:
x+x/2!+x/3!+x/4!+...+x/n!
To calculate and print the sum of the following series:
x/2!+x2/3!+x3/4!+x4/5!+...+xn/(n+1)!
To calculate and print the sum of the following series:
x/2! - x2/3!+x3/4! - x4/5!+...±xn/(n+1)!
where the symbol ! stands for factorial eg. 5!=5*4*3*2*1, 3!=3*2*1

Ans.

```
import java.util.*;
class Series
{
   double x;
   int n;
   Series(double x,int n)
   {
      this.x=x;
      this.n=n;
   }

   void sumSeries1()
   {
      double s=0;
      long i,p,j;
      for(i=1;i<=n;i++)
      {
         p=1;
         for(j=1;j<=i;j++)
         {
            p=p*j;
         }
         s=s+x/p;
      }
      System.out.println("Sum="+s);
   }
   void sumSeries2()
   {
      double s=0;
      long i,p,j;
      for(i=1;i<=n;i++)
```

```
        {
            p=1;
            for(j=1;j<=i+1;j++)
            {
                p=p*j;
            }
            s=s+x*i/p;
        }
        System.out.println("Sum="+s);
    }
    void sumSeries3()
    {
        double s=0;
        long i,p,j;
        for(i=1;i<=n;i++)
        {
            p=1;
            for(j=1;j<=i+1;j++)
            {
                p=p*j;
            }
            if(i%2!=0)
                s=s+x*i/p;
            else
                s=s-x*i/p;
        }
        System.out.println("Sum="+s);
    }
}
```

20. Create a class named Rounding which contains the following members:

Data members:
n and r of double type
Member Functions:
i. Constructor to read a real number from the keyboard into n.
ii. to round off n to two places of decimal and store it in r.
iii. to display the value of the data members.

Ans.

```
import java.util.*;
class Rounding
{
    double n,r;
```

```
    Rounding()
    {
       Scanner sc=new Scanner(System.in);
       System.out.println("Enter a real number in n:");
       n=sc.nextDouble();
    }

    void roundOff()
    {
       r=Math.round(n*100)/100.0;
    }
    void display()
    {
       System.out.println("n="+n+"r="+r);
    }
}
```

Chapter 6
Library Classes

Mental Drill

Tick (✓) the correct option.

1. Which of the following is a composite data type?
   a. int
   b. String
   c. char
   d. float

   Ans. b. String

2. Which of the following is not a wrapper class?
   a. Byte
   b. Int
   c. Long
   d. Float

   Ans. b. Int

3. Which of the following is valid method of initialising?
   a. Integer a=new (5);
   b. Integer a=Integer(5);
   c. Integer a=new Integer(5);
   d. Integer a=new (“5”);

   Ans. d. Integer a=new (“5”);

4. If s=“123”, which among the following will convert it to an integer?
   a. int a=Integer(s);
   b. int a=(int)s;
   c. int a=parseInt(s);
   d. int a=Integer.parseInt(a);

   Ans. d. int a=Integer.parseInt(a);

5. Which among the following is valid character initialisation?
   a. Character c=new Character(‘c’);
   b. Character c=new Character(“c”);
   c. Both a and b
   d. None of these

   Ans. a. Character c=new Character(‘c’);

6. Which among the following methods return a true or a false?
   a. toLowerCase()
   b. toUpperCase()
   c. parseInt()
   d. isUpperCase()

   Ans. d. isUpperCase()

7. What package is a part of the wrapper class which is imported by default into all Java programs?
   a. java.util
   b. java.lang
   c. java.awt
   d. None of these

   Ans. b. java.lang

8. Which class is inherited by the Character and Boolean wrapper classes?

a. Object
b. Number
c. Both a and b
d. None of these

Ans. a. Object

9. What is, converting a primitive value into an object of the corresponding wrapper class called?

a. Autoboxing
b. Unboxing
c. Type Casting
d. All of these

Ans. a. Autoboxing

10. Which among the following function is used to check whether a given character is a tab space or not?

a. isTab()
b. isTabSpace()
c. isSpace()
d. isWhitespace()

Ans. d. isWhitespace()

SECTION A

A. Answer the following questions.

1. What is composite data type?

Ans. In computer science, a composite data type or compound data type is any data type which can be constructed in a program using the programming language's primitive data types and other composite types.

2. State the difference between primitive and composite data type.

Ans.

Primitive data type	Composite data type
Primitives are the fundamental data types that are represented by keywords namely byte, short, int, long, char and boolean.	Composite data types are user defined types and may have any name defined by the user.
Variables of primitive data type stores values of corresponding data type.	Variables of composite data type stores reference to an object.
It can store only one type of value.	It can store data of multiple data type.
The size is predetermined.	The size is not predetermined.

3. Explain with an example, how objects are passed as a reference.

Ans. Example of how objects are passed as reference:

```
class PassByReference
{
    static void change(Number r)
    {
        r.a=r.a+5;
        r.b=r.b+5;
    }
        static void call()
    {
        Number n=new Number(5,6);
        System.out.println(n.a+"\t"+n.b);
        change(n);
        System.out.println(n.a+"\t"+n.b);
    }
}
```

will result in the following output when the call( ) function is executed:

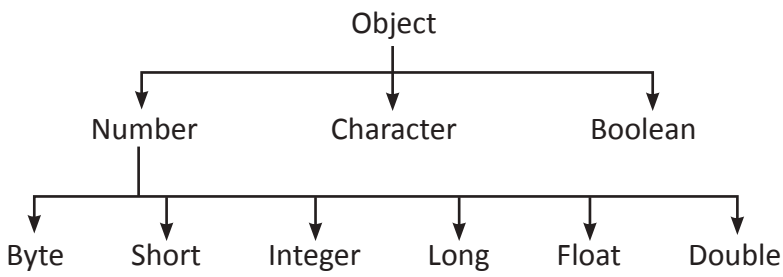
```
5   6
10  1
```

4. What is a wrapper class? Name the wrapper classes used in java.

Ans. A Wrapper class is a class whose object wraps or contains a primitive data types. Wrapper classes used in Java- Byte, Short, Integer, Long, Character, Boolean, Float and Double.

5. Draw the wrapper class hierarchy.

Ans.



6. Show with an example, how wrapper class object can be created and initialised?

Ans. Example:

```
Byte b=new Byte(byte) 10):
```

7. Explain autoboxing and unboxing in java.

Ans. Converting a primitive value into an object of the corresponding wrapper class is called autoboxing. Converting an object of a wrapper type to its corresponding primitive value is called unboxing.

8. Explain the usage of <Wrapper Class>.parse…( ) method.

Ans. This function is used to convert a string to corresponding primitive data type. This function is present in all wrapper classes except Character and Boolean wrapper classes. The following example shows how this function can be used.

```
byte b=Byte.parseByte("12");
short s=Short.parseShort("12");
int i=Integer.parseInt("123");
long l=Long.parseLong("98412");
float f=Float.parseFloat("12.3f");
double d=Double.parseDouble("12.46");
```

9. What is the difference between isUpperCase() and toUpperCase() method?

Ans. The isUpperCase() method is used to check whether a given character is in uppercase or not. The toUpperCase() method is used to convert a given character to uppercase.

10. What is the use of isWhitespace() method in Java?

Ans. A whitespace in Java is used as a space, tab, or a new line, and this method determines whether the given char(ch) is a whitespace or not.

11. Why is a class known as a composite data type?

Ans. A composite data type is one which is composed with various primitive data type. A class defined with various primitive data types such as int, double etc; so it is known as a composite data type; and it is used to create objects which hold similar types of values and behaviours (functions).

12. Name the package that contain the wrapper classes.

Ans. java.lang

B. Short answer type questions.

1. Answer as directed:

a. Assign a integer 123 to an int variable a.

b. Convert the variable a to a String and store it in the variable s.

Ans. a. int a=123;

b. String s=Integer.toString(a);

2. Answer as directed:

a. Assign an string "123" to a String variable a.

b. Convert the variable a to an integer and store it in the variable s.

Ans. a. String a="123";

b. int s=Integer.parseInt(a);

3. For the program given below mark the parts ___1___ and __2__ as autoboxing or unboxing.

```
class Example
{
        static void main()
```

```
        {
            Integer i;
            int sum=0;
            for(i=1;i<=10;i++) //___1___
            {
                sum=sum+i; //__2__
            }
            System.out.println(sum);
        }
}
```

Ans. __1__:Autoboxing

__2__:Unboxing

4. Give the output of the following program when the main( ) method is executed:

```
class Overload
{
        static void wrapper(int a)
        {
            System.out.println("int type="+a);
        }
        static void wrapper(Integer a)
        {
            System.out.println("Integer type="+a);
        }
        static void main()
        {
            int b=13;
            Integer c=new Integer(b);
            wrapper(c);
            wrapper(b);
        }
}
```

Ans. Output:

```
Integer type=13
int type=13
```

5. Write one word answer for the following:

(i) A method that converts a string to a primitive integer data type

(ii) The default initial value of a boolean variable data type

Ans. (i) Integer.parseInt()

(ii) false

6. Write the output of the following:

(i) System.out.println (Character.isUpperCase('R'));

(ii) System.out.println(Character.toUpperCase('j'));

Ans. (i) true

(ii) J

7. State the method that:

(i) Converts a string to a primitive float data type

(ii) Determines if the specified character is an uppercase character

Ans. (i) Float.parseFloat()

(ii) Character.isUpperCase()

8. State the data type and the value of y after the following is executed:

```
char x ='7';
y = Character.isLetter(x);
```

Ans. y is of boolean type its value is false.

9. Write the return type of the following library functions:

(i) isLetterOrDigit(char)

(ii) parseInt()

Ans. (i) boolean

(ii) int

10. State the data type and value of res after the following is executed:

```
char ch='t';
res=Character.toUpperCase(ch);
```

Ans. res is of char type is value is 'T'.

11. Give the output of the following code:

```
String A="26", B="100";
String D=A+B+"200";
int x=Integer.parseInt(A);
int y=Integer.parseInt(B);
int d=x+y;
System.out.println("Result 1="+D);
System.out.println("Result 2="+d);
```

Ans. Output:

```
Result 1= 26100200
Result 2=126
```

12. Write a difference between the functions isUpperCase() and toUpperCase().

Ans. Difference:

isUpperCase()	toUpperCase()
Checks whether a given character is in uppercase or not.	Converts a given character to uppercase.
Return type is boolean.	Return type is char data type.

SECTION B

Write programs for the following:

1. **Input a character and check whether it is an alphabet or not, if it is check whether it is in uppercase or in lower case.**

   Example:
   INPUT: A
   Output: Uppercase
   Input: 2
   Output: Not an alphabet

Ans.

```
import java.util.*;
class Sol1
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      char c;
      System.out.println("Enter a character:");
      c=sc.next().charAt(0);
      if (Character.isLetter(c))
      {
         if (Character.isUpperCase(c))
            System.out.println("Uppercase");
         else
            System.out.println("Lowercase");
      }
      else
         System.out.println("Not an alphabet");
   }
}
```

2. **Create a function int sum(integer a) which accepts an Integer object as parameter and return the sum of its digits. In the main() input an integer and find the sum of its digits using the above method.**

Ans.

```
import java.util.*;
class Sol2
{
   static int sum(Integer a)
   {
      int d,s=0;
      while(a>0)
```

```
    {
      d=a%10;
      s+=d;
      a/=10;
    }
    return s;
  }

  static void main()
  {
    Scanner sc=new Scanner(System.in);
    Integer n,s;
    System.out.println("Enter a Number:");
    n=sc.nextInt();
    s=sum(n);
    System.out.println("Sum:"+s);
  }
}
```

3. Create a function Boolean isPalindrome(Integer a) which checks whether the Integer object 'a' is a palindrome number or not. In the main input into an int type variable and check whether it is a palindrome number or not. Palindrome number is such a number which is same as its reverse,

example 121, 1331 etc.

Ans.

```
import java.util.*;
class Sol3
{
  static Boolean isPalindrome(Integer a)
  {
    int i,c=0;
    for(i=1;i<=a;i++)
    {
      if(a%i==0)
        c++;
    }
    if(c==2)
      return true;
    else
      return false;
  }

  static void main()
  {
    Scanner sc=new Scanner(System.in);
```

```
        Integer n;
        boolean f;
        System.out.println("Enter a Number:");
        n=sc.nextInt();
        f=isPalindrome(n);
        if(f)
          System.out.println("Palindrome");
        else
          System.out.println("Not Palindrome");
    }
}
```

4. Input 10 characters and find the frequency of uppercase and lowercase characters separately.

For example:
Input: A, b, c, d, E, F, G, H, I, j
Ouput: Uppercase Frequency: 6
Lowecase Frequency: 4

Ans.

```
import java.util.*;
class Sol4
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i,u=0,l=0;
      char c;
      System.out.println("Enter 10 characters:");
      for(i=1;i<=10;i++)
      {
         c=sc.next().charAt(0);
         if(Character.isUpperCase(c))
            u++;
         else if(Character.isLowerCase(c))
            l++;
      }
      System.out.println("Uppercase Frequency:"+u);
      System.out.println("Lowercase Frequency:"+l);
   }
}
```

5. Input 10 characters and concatenate only those characters which are either a letter or a digit.

For example:

Input: A, #, 1, d, E, F, %, H, 5, j

Ouput: A1dEFH5

Ans.

```
import java.util.*;
class Sol5
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int i;
      char c;
      String s=" ";
      System.out.println("Enter 10 characters:");
      for(i=1;i<=10;i++)
      {
         c=sc.next().charAt(0);
         if(Character.isLetterOrDigit(c))
            s+=c;
      }
      System.out.println("Result:"+s);
   }
}
```

6. Input an integer and form a new number by removing all odd digits from it.

For example,
Input: 123456
Output: 246

Ans.

```
import java.util.*;
class Sol6
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int n,d,i,nn;
      String s="";
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=n;i>0;i/=10)
      {
         d=i%10;
         if(d%2==0)
            s=d+s;
      }
      nn=Integer.parseInt(s);
      System.out.println("Result:"+nn);
   }
}
```

7. Input an integer and remove all even digits and display it. Now check whether the new number is a perfect number or not.

Ans.

```
import java.util.*;
class Sol7
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int n,d,i,nn,sum=0;
      String s="";
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=n;i>0;i/=10)
      {
         d=i%10;
         if(d%2!=0)
            s=d+s;
      }
      nn=Integer.parseInt(s);
      for(i=1;i<nn;i++)
      {
         if(nn%i==0)
            sum+=i;
      }
      if(sum==nn)
         System.out.println("Perfect Number");
      else
         System.out.println("Not a Perfect Number");
   }
}
```

8. Input 10 characters and form a new string by concatenating all the characters by converting all uppercase to lowercase and vice-versa.

For example:
Input: A, b, 1, d, E, F, G, H, @, #
Ouput: aB1Defgh@#

Ans.

```
import java.util.*;
class Sol8
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
```

```
        int i;
        char c;
        String ns="";
        System.out.println("Enter 10 characters:");
        for(i=1;i<=10;i++)
        {
              c=sc.next().charAt(0);
              if(Character.isUpperCase(c))
              c=Character.toLowerCase(c);
              else if(Character.isLowerCase(c))
              c=Character.toUpperCase(c);
              ns=ns+c;
        }
        System.out.println("Result:"+ns);
    }
}
```

9. Design a class to overload a function sum() as follows:

(a) void sum(int n) – with one int argument that finds the sum of the digits in n.
(b) void sum(Integer n) – with one Integer argument that finds the sum of the digits in n.
Also create the main() method to call the above methods.

Ans.

```
class Overload
{
   void sum(int n)
   {
      int d,s=0;
      while(n>0)
      {
         d=n%10;
         s+=d;
         n/=10;
      }
      System.out.println("Sum="+s);
   }
   void sum(Integer n)
   {
      Integer d,s=0;
      while(n>0)
      {
         d=n%10;
         s+=d;
         n/=10;
      }
      System.out.println("Sum="+s);
```

```
    }
    void main()
    {
       int a=1234;
       Integer b=23498;
       sum(a);
       sum(b);
    }
}
```

10. Create a class called Number having the following members:

Instance Variables:
a and b of int type
Member Functions:
- **Parameterised constructor to initailise a and b.**
- **void swap() that interchanges the value of a and b.**
- **void display() to display the data members.**

In the main() method input 2 interchange their values using the methods of the above class.

Ans.

```
import java.util.*;
class Number
{
   int a,b;
   Number(int x,int y)
   {
      a=x;
      b=y;
   }
   void swap()
   {
      int t=a;
      a=b;
      b=t;
   }
   void display()
   {
      System.out.println(a+" "+b);
   }


   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int x,y;
      System.out.println("Enter 2 numbers:");
```

```
        x=sc.nextInt();
        y=sc.nextInt();
        Number ob=new Number(x,y);
        ob.swap();
        ob.display();
    }
}
```

11. Create a class called StringNumber having the following members:

Instance Variables:
s of String type
n of int type
Member functions:

- **StringNumber(String t) – to initialize s with t that contains a number (example "123") and n with 0.**
- **void assign() – assign the number stored in s to n after converting it to an integer.**
- **void largest() – find the largest digit in n and display it.**
  Also create the main to show the implementation.

Ans.

```
import java.util.*;
class StringNumber
{
    String s;
    int n;
    StringNumber(String t)
    {
        s=t;
        n=0;
    }
    void assign()
    {
        n=Integer.parseInt(s);
    }
    void largest()
    {
        int i,d,l=0;
        for(i=n;i>0;i/=10)
        {
            d=i%10;
            if(l==0)
            l=d;
            if(d>l)
            l=d;
        }
        System.out.println("Largest Digit:"+l);
```

```
    }

    static void main()
    {
        Scanner sc=new Scanner(System.in);
        String s;
        System.out.println("Enter a number:");
        s=sc.nextLine();
        StringNumber ob=new StringNumber(s);
        ob.assign();
        ob.largest();
    }
}
```

12. Create a function Boolean isBouncy(Integer n) to check whether it is a Bouncy Number or not.

Increasing Number : Working from left-to-right if no digit is exceeded by the digit to its left it is called an increasing number; for example, 22344.

Decreasing Number : Similarly if no digit is exceeded by the digit to its right it is called a decreasing number; for example, 774410.

Bouncy Number : We shall call a positive integer that is neither increasing nor decreasing a "bouncy" number; for example, 155349. Clearly there cannot be any bouncy numbers below 100.

Ans.

```
class Sol12
{
    Boolean isBouncy(Integer n)
    {
        int d,i,fa=0,fd=0,sd;
        for(i=n;i>9;i/=10)
        {
            d=i%10;
            sd=(i/10)%10;
            if(sd>d)
                fa=1;
            if(sd<d)
                fd=1;
        }
        if(fa==1 && fd==1)
            return true;
        else
            return false;
    }
}
```

13. Design a program to accept a positive whole number and find the binary equivalent of the number and count the number of 1's in it and display whether it is a Evil number or not with an appropriate message. Output the result in format given below:

Example 1
INPUT : 15
BINARY EQUIVALENT : 1111
NO. OF 1's : 4
OUTPUT : EVIL NUMBER
Example 2
INPUT : 26
BINARY EQUIVALENT : 11010
NO. OF 1's : 3
OUTPUT : NOT AN EVIL NUMBER

Ans.

```
import java.util.*;
class Sol13
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int n,d,i,c=0;
      String s=" ";
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=n;i>0;i/=2)
      {
         d=i%2;
         if(d==1)
            c++;
         s=d+s;
      }
      System.out.println("Binary Equivalent:"+s);
      System.out.println("No. of 1's:"+c);
      if(c%2==0)
         System.out.println("Evil Number");
      else
         System.out.println("Not an Evil Number");
   }
}
```

14. Write a Program in Java to input a number and check whether it is a Fascinating Number or not.

Fascinating Numbers : Some numbers of 3 digits or more exhibit a very interesting property. The property is such that, when the number is multiplied by 2 and 3, and both these products are concatenated with the original number, all digits from 1 to 9 are present exactly once, regardless of the number of zeroes.

Let's understand the concept of Fascinating Number through the following example:

Consider the number 192,

192 x 1 = 192

192 x 2 = 384

192 x 3 = 576

Concatenating the results : 192384576

It could be observed that '192384576' consists of all digits from 1 to 9 exactly once. Hence, it could be concluded that 192 is a Fascinating Number.

Some examples of fascinating Numbers are : 192, 219, 273, 327, 1902, 1920, 2019, etc.

Ans.

```
import java.util.*;
class Sol14
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      long n,d,i,t,nn,f=0,j,c;
      String s="";
      System.out.println("Enter a number:");
      n=sc.nextInt();
      for(i=1;i<=3;i++)
      {
         t=n*i;
         s=s+t;
      }
      nn=Long.parseLong(s);
      for(i=1;i<=9;i++)
      {
         c=0;
         for(j=nn;j>0;j/=10)
         {
            d=j%10;
            if(d==i)
               c++;
         }
```

```
        if(c!=1)
          f=1;
      }
      if(f==0)
        System.out.println("Fascinating number");
      else
        System.out.println("Not a Fascinating number");
    }
}
```

15. Write a program to accept a number and check and display whether it is a Niven number of not.

(Niven number is that number which is divisible by its sum of digits).
Example:
Consider the number 126.
Sum of its digits is 1 + 2 + 6 = 9 and 126 is divisible by 9.

Ans.

```
import java.util.*;
class Sol15
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int n,d,i,s=0;
    System.out.println("Enter a number:");
    n=sc.nextInt();
    for(i=n;i>0;i/=10)
    {
      d=i%10;
      s=d+s;
    }
    if(n%s==0)
      System.out.println("Niven Number");
    else
      System.out.println("Not a Niven Number");
  }
}
```

Chapter 7
Encapsulation

Mental Drill

Tick (✓) the correct option.

1. Which access specifier allows accessibility of a member only within the same class where it is declared?

   a. public
   b. private
   c. protected
   d. default

   Ans. b. private

2. Wrapping up of data and method into a single unit is called _____.

   a. Encapsulation
   b. Abstraction
   c. Inheritance
   d. Polymorphism

   Ans. a. Encapsulation

3. An OOP feature that allows all members of a class to be the members of another class.

   a. Encapsulation
   b. Abstraction
   c. Inheritance
   d. Polymorphism

   Ans. c. Inheritance

4. Which keyword allows Inheritance?

   a. extend
   b. extends
   c. for
   d. protected

   Ans. b. private

5. Which access specifier allows accessibility by all classes in the same package, but only by subclasses in a different package?

   a. public
   b. private
   c. protected
   d. default

   Ans. c. protected

6. A static member a is declared in a class named 'Myclass', write the statement to initialise it with 5, from a function in another class.

   a. a=5;
   b. a of Myclass=5;
   c. Myclass.a=5;
   d. a.Myclass=5;

   Ans. c. Myclass.a=5;

7. Which among the following best describes encapsulation?

   a. It is a way of combining various data members into a single unit

   b. It is a way of combining various member functions into a single unit

   c. It is a way of combining various data members and member functions into a single unit which can operate on any data

   d. It is a way of combining various data members and member functions that operate on those data members into a single unit

Ans. d. It is a way of combining various data members and member functions that operate on those data members into a single unit

8. If data members are private, what can we do to access them from the class object?

   a. Create public member functions to access those data members

   b. Create private member functions to access those data members

   c. Create protected member functions to access those data members

   d. Private data members can never be accessed from outside the class

Ans. a. Create public member functions to access those data members

9. Which feature can be implemented using encapsulation?

   a. Inheritance

   b. Abstraction

   c. Polymorphism

   d. Overloading

Ans. b. Abstraction

10. How can Encapsulation be achieved?

    a. Using Access Specifiers

    b. Using only private members

    c. Using inheritance

    d. Using Abstraction

Ans. a. Using Access Specifiers

Section A

Answer the following questions.

1. What is encapsulation?

Ans. Wrapping up of data and methods into a single unit is called encapsulation.

2. What are access specifiers used for? Name the access specifiers used in Java.

Ans. Access modifiers (or access specifiers) are keywords in object-oriented languages that set the accessibility of classes, methods, and other members. Access modifiers are a specific part of programming language syntax used to facilitate the encapsulation of components.

Access specifiers used in Java are:

- Default
- private
- protected
- public

3. State the difference between:

a. Public and private access specifier

b. Protected and default access specifier

Ans. a. Public access specifier allows accessibility inside a package as well as outside the package. Private access specifier allows accessibility only within the class.

b. The protected specifier allows access by all subclasses of the class in a program, whatever package they reside in, as well as to other code in the same package. The default specifier allows access by other code in the same package, but not by code that is in subclasses residing in different packages.

4. What can't a class be declared as private?

Ans. A top-level class as private would be completely useless because nothing would have access to it.

5. Give an example to illustrate overloaded constructor with different access specifier.

Ans. Program to show how overloaded constructors can have different access specifiers and how are they accessed from a different class:

```
class Access
{
        private int a,b;
        private Access(int x,int y)
        //constructor with private access specifier
        {
                a=x;
                b=y;
        }
        Access(int x) //overloaded constructor with default access.
        {
                a=b=x;
        }
        void show( )
        {
                System.out.println(a+ " "+b);
        }
        static void createObject( )
        {
                Access obj=new Access(5,6);
                obj.show();
```

```
        }
}
    public class MainClass
    {
        public static void main(String args[ ])
        {
            /*Access obj=new Access(5,6);Error constructor has private
            access specifier.*/
            Access.createObject();
            Access obj=new Access(7); /*Correct as the default access
            version of the constructor is being called.*/
        }
    }
```

6. What is Inheritance? Which keyword is used to implement inheritance?

Ans. Inheritance is a mechanism wherein a new class is derived from an existing class. The 'extends' keyword is used to implement inheritance.

7. What is a package? What is its significance?

Ans. A package is a namespace that organises a set of related pre-compiled classes and interfaces into folders.

Packages are used for:

- Preventing naming conflicts.
- Making searching/locating and usage of classes, interfaces, enumerations and annotations easier
- Providing controlled access: protected and default have package level access control.
- Packages can be considered as data encapsulation (or data-hiding).

8. How are packages created in BlueJ?

Ans. To create a package:

Step 1: Select the Edit menu and select the New Package… option.

Step 2: Enter the name of the package in the 'Create New Package' dialog and hit the 'OK' button.

9. What do you understand by "Scope of a Variable"?

Ans. The scope of a variable is defined as the extent of the program code within which the variable can be accessed or declared or worked with.

10. What is the scope of argument variables?

Ans. Argument variables have their scope limited only within the method or constructor block where it is declared.

SECTION B

Write programs for the following:

1. Create a class name Check having the following:

private data members: c of char type.
public member functions:
i. Parameterised constructor to initialise it with a character.
ii. To return true if it is in uppercase else return false.
Also create a main() and call the member methods.

Ans.

```
import java.util.*;
class Check
{
   private char c;
   public Check(char x)
   {
      c=x;
   }
  public  boolean check()
   {
      if(c>='A' && c<='Z')
         return true;
      else
         return false;
   }

   public static void main(String args[])
   {
      char x;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a character:");
      x=sc.next().charAt(0);
      Check ob=new Check(x);
      if(ob.check()==true)
         System.out.println("Uppercase");
      else
         System.out.println("Not in upper case");
   }
}
```

2. Define a class Candidate with the following descriptions:

Private Members:

- A data member RNo(Registration Number) of type long
- A data member Name of type String
- A data member Score of type float
- A data member Remarks of type String
- A member function AssignRem() to assign the remarks as per the score obtained by a candidate. Score range and the respective remarks are shown as follows:

Score	Remarks
>=50	Selected
Less than 50	Not Selected

Public Members:

- A function ENTER() to allow user to enter values for Rno, Name, Score and call function AssignRem() to assign grade.
- A function DISPLAY() to allow user to view the content of all data members.
  Also create a main() method to create an object and show its implementation by calling the above methods.

Ans.

```
import java.util.*;
class Candidate
{
   private long Rno;
   private String Name;
   private float Score;
   private String Remarks;
   private void AssignRem()
   {
        if(Score>=50)
        Remarks="Selected";
        else
        Remarks="Not Selected";
   }
   public void ENTER()
   {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the reg. no.:");
        Rno=sc.nextLong();
        System.out.println("Enter the name:");
        Name=sc.nextLine();
        System.out.println("Enter the score:");
        Score=sc.nextFloat();
        AssignRem();
   }
```

```
    public void DISPLAY()
    {
      System.out.println("Reg. no.:"+Rno);
      System.out.println("Name:"+Name);
      System.out.println("Score:"+Score);
      System.out.println("Remarks:"+Remarks);
    }

    public static void main(String args[])
    {
      Candidate ob=new Candidate();
      ob.ENTER();
      ob.DISPLAY();
    }
}
```

3. Define a function RESTRA with the following descriptions:

Private members:

- FoodCode of type int
- Food of type String
- FType of type String
- Sticker of type String
- A member function GetSticker() to assign the following values for Sticker as per the given FType:

FType	Sticker
Sticker	GREEN
Contains Egg	YELLOW
Non-Vegetarian	RED

Public Members:

- A function GetFood() to allow user to enter values for FoodCode, Food, FType and call function GetSticker() to assign Sticker.
- A function ShowFood() to allow user to view the content of all the data members.
  Also create a main() method to create an object and show its implementation by calling the above methods.

Ans.

```
import java.util.*;
class RESTRA
{
  private int FoodCode;
  private String Food,FType,Sticker;
  private void GetSticker()
  {
```

```
        if(FType.equalsIgnoreCase("Sticker"))
          Sticker="GREEN";
        else if(FType.equalsIgnoreCase("Contains Egg"))
          Sticker="YELLOW";
        else if(FType.equalsIgnoreCase("Non-Vegetarian"))
          Sticker="YELLOW";
    }
    public void GetFood()
    {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the FoodCode:");
      FoodCode=sc.nextInt();
      System.out.println("Enter the Food:");
      Food=sc.nextLine();
      System.out.println("Enter the Food Type:");
      FType=sc.nextLine();
      GetSticker();
    }
    public void ShowFood()
    {
      System.out.println("FoodCode:"+FoodCode);
      System.out.println("Food:"+Food);
      System.out.println("Food Type:"+FType);
      System.out.println("Sticker:"+Sticker);
    }

    public static void main(String args[])
    {
      RESTRA ob=new RESTRA();
      ob.GetFood();
      ob.ShowFood();
    }
}
```

4. Define a class Seminar with the following specification:

Private members:

- SeminarI long
- Topic String
- VenueLocation String
- Fee float
- CalcFee() function to calculate Fee depending on VenueLocation

VenueLocation	Fee
Outdoor	5000
Indoor Non-AC	6500
Indoor AC	7500

Public members:

- Register() function to accept values for SeminarID, Topic, VenueLocation and call CalcFee() to calculate Fee.
- ViewSeminar() function to display all the data members on the screen.
  Also create a main() method to create an object and show its implementation by calling the above methods.

Ans.

```
import java.util.*;
class Seminar
{
   private long SeminarID;
   private String Topic,VenueLocation;
   private float Fee;
   private void CalcFee()
   {
      if(VenueLocation.equalsIgnoreCase("Outdoor"))
         Fee=5000;
      else if(VenueLocation.equalsIgnoreCase("Indoor Non-AC"))
         Fee=6500;
      else if(VenueLocation.equalsIgnoreCase("Indoor AC"))
         Fee=7500;
   }
   public void Register()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the Seminar ID:");
      SeminarID=sc.nextInt();
      System.out.println("Enter the Topic:");
      Topic=sc.nextLine();
      System.out.println("Enter the Venue Location:");
      VenueLocation=sc.nextLine();
      CalcFee();
   }
   public void ViewSeminar()
   {
      System.out.println("Seminar ID:"+SeminarID);
      System.out.println("Topic:"+Topic);
      System.out.println("Venue Location:"+VenueLocation);
      System.out.println("Fee:"+Fee);
   }

   public static void main(String args[])
   {
      Seminar ob=new Seminar();
```

```
        ob.Register();
        ob.ViewSeminar();
    }
}
```

5. Define a class Graments with the following description:

Private members:

- GCode of type String
- GType of type String
- GSize of type integer
- GFabric of type String
- GPrice of type String
- A function Assign() which calculates and assigns the value of GPrice as follows:

For the GFabric as "COTTON"

GType	GPrice
TROUSER	1300
SHIRT	1100

For GFabric other than "COTTON" the above mentioned GPrice gets reduced by 10%.

Public Members:

- A constructor to assign initial values of GCode, GType and GFabric with the words "NOT ALLOTED" and GSize and GPrice with 0.
- A function input() to input the values of the data members of GCode, GType, GFabric, GSize and invoke the Assign function.
- A function Display() which displays the content of all the data members for a Garment. Also create a main() method to create an object and show its implementation by calling the above methods.

Ans.

```
import java.util.*;
class Garments
{
   private String GCode,GType,GFabric;
   private int GSize;
   private float GPrice;
   private void  Assign()
   {
      if(GType.equalsIgnoreCase("TROUSER"))
         GPrice=1300;
      else if(GType.equalsIgnoreCase("SHIRT"))
         GPrice=1100;

      if(!(GFabric.equalsIgnoreCase("COTTON")))
         GPrice=GPrice-10/100f*GPrice;
   }
   public Garments()
```

```
    {
       GCode=GType=GFabric=”NOT ALLOTED”;
       GPrice=GSize=0;
    }
    public void input()
    {
      Scanner sc=new Scanner(System.in);
       System.out.println(“Enter the GCode:”);
       GCode=sc.nextLine();
       System.out.println(“Enter the GType:”);
       GType=sc.nextLine();
       System.out.println(“Enter the GFabric:”);
       GFabric=sc.nextLine();
       System.out.println(“Enter the GSize:”);
       GSize=sc.nextInt();
       Assign();
    }
    public void Display()
    {
       System.out.println(“GCode:”+GCode);
       System.out.println(“GType:”+GType);
       System.out.println(“GFabric:”+GFabric);
       System.out.println(“GSize:”+GSize);
       System.out.println(“GPrice:”+GPrice);
    }

    public static void main(String args[])
    {
       Garments ob=new Garments();
       ob.input();
       ob.Display();
    }
}
```

6. Define a class Bus with the following specifications:

   Private Members:

   - BusNo - to store Bus Number
   - From – to store place name of origin
   - To – to store place name of the destination
   - Type – to store Bus Type ‘O’ for ordinary, ‘E’ for Economy and ‘L’ for luxury
   - Distance – to store the distance in kilometres
   - Fare - to store the bus fare
   - A function CalcFare() to calculate Fare as per the following criteria:

Type	Fare
'O'	15*Distance
'E'	20*Distance
'L'	24*Distance

Public Members:

- A constructor function to initialise Type as 'O' and Fare as 500.
- A function Allocate() to allow user to enter values for BusNo, From, To, Type and Distance. Also this function should call CalcFare() to calculate Fare.
- A function Show() to display the content of all the data members on the screen. Also create a main() method to create an object and show its implementation by calling the above methods.

Ans.

```
import java.util.*;
class Bus
{
   private String From,To;
   private char Type;
   private int BusNo;
   private float Distance,Fare;
   private void  CalcFare()
   {
      if(Type=='O')
         Fare=15*Distance;
      else if(Type=='E')
         Fare=20*Distance;
      else if(Type=='L')
         Fare=24*Distance;
   }
   public Bus()
   {
      Type='O';
      Fare=500;
   }
   public void Allocate()
   {
     Scanner sc=new Scanner(System.in);
      System.out.println("Enter the bus no.:");
      BusNo=sc.nextInt();
      System.out.println("Enter the origin:");
      From=sc.nextLine();
      System.out.println("Enter the destination:");
      To=sc.nextLine();
      System.out.println("Enter the Type of bus:");
```

```
        Type=sc.next().charAt(0);
        System.out.println("Enter the Distance:");
        Distance=sc.nextFloat();
        CalcFare();
    }
    public void Show()
    {
        System.out.println("the bus no.:"+BusNo);
        System.out.println("the origin:"+From);
        System.out.println("the destination:"+To);
        System.out.println("Type of bus:"+Type);
        System.out.println("Distance:"+Distance);
        System.out.println("Fare:"+Fare);
    }

    public static void main(String args[])
    {
        Bus ob=new Bus();
        ob.Allocate();
        ob.Show();
    }
}
```

7. Design a class CABS with the following specifications:

Private Members:

- CNo - to store Cab Number
- Type - to store a character 'A', 'B' or 'C' as city type
- PKM - to store per Kilometer charges
- Dist - to store Distance travelled (in KM)

Public Members:

- A constructor function to initialise Type as 'A' and CNo as '1111'.
- A function Charges() to assign PKM as per the following table:

Type	PKM
A	25
B	C
'L'	15

- A function Register() to allow administrator to enter the values for CNo and Type. Also, this function should call Charges() for PKM charges.
- A function ShowCab() to allow user to the Distance and display CNo, Type, PKM, PKM*Distance (as Amount) on screen.

  Create another class named Taxi in the same program to create the main() in which an object of the CABS class is to be created and the relevant member functions of it should be called to show its implementation.

Ans.

```
import java.util.*;
class CABS
{
   private char Type;
   private int CNo;
   private float PKM,Dist;
   public CABS()
   {
      Type='A';
      CNo=1111;
   }
   private void  Charges()
   {
      if(Type=='A')
         PKM=25;
      else if(Type=='B')
         PKM=20;
      else if(Type=='L')
         PKM=15;
  }

   public void Register()
   {
     Scanner sc=new Scanner(System.in);
      System.out.println("Enter the cab no.:");
      CNo=sc.nextInt();
      System.out.println("Enter the type:");
      Type=sc.next().charAt(0);
      Charges();
   }
   public void ShowCab()
   {
      Scanner sc=new Scanner(System.in);
      float amount;
      System.out.println("Enter the cab no.distance travelled:");
      Dist=sc.nextFloat();
      amount=PKM*Dist;
      System.out.println("the cab no.:"+CNo);
      System.out.println("the type:"+Type);
      System.out.println("PKM:"+PKM);
      System.out.println("Amount:"+amount);
    }
```

```
    public static void main(String args[])
    {
      CABS ob=new CABS();
      ob.Register();
      ob.ShowCab();
    }
}
```

8. The following question consists of 2 parts, go through it carefully and answer them:

(i) Consider a package named "Number" exist, in which you need to create a class with adequate access specifier with the following specifications.

Class Name: Armstrong
Private Member:
N of int type
Public Member:

- Parameterised constructor to initialise N.
- Function isArmstrong() that checks whether the number in N is an Armstrong number or not and accordingly returns a true or a false.

(ii) Outside the package using suitable import statements create a class called "Check" that will have a main() function defined where an integer is taken from the user using Scanner and using the member functions of the "Number" class check whether it is an Armstrong number or not.

Ans.

(i)

```
package Number;
public class Armstrong
{
  private int N;
  public Armstrong(int n)
  {
    N=n;
  }
  public boolean isArmstrong()
  {
    int i,d,s=0;
    for(i=N;i>0;i/=10)
    {
      d=i%10;
      s+=d*d*d;
    }
    if (s==N)
      return true;
    else
      return false;
```

```
    }
}
```

(ii)

```
import Number.Armstrong;
import java.util.Scanner;
class Check
{
   public static void main(String args[])
   {
      Scanner sc=new Scanner(System.in);
      int n;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      Armstrong ob=new Armstrong(n);
      if (ob.isArmstrong())
         System.out.println("Armstrong No.");
      else
         System.out.println("Not an Armstrong No.");
      }
   }
```

9. Define a class Employee having the following description:-

Private Members:

o int pan	to store personal account number
o String name	to store name
o double taxincome	to store annual taxable income.
o double tax	to store tax that is calculated

Public Members:

- input() : Store the pan number, name, taxeincome
- calc() : Calculate tax for an employee
- display() : Output details of an employee i.e. display all the data members.

Write a program to compute the tax according to the given conditions and display the output as per given format.

Total Annual Taxable Income	**Tax Rate**
Upto ₹. 1,00,000	No tax
From 1,00,001 to 1,50,000	10% of the income exceeding ₹. 1,00,000
From 1,50,001 to 2,50,000	₹. 5000+20% of the income exceeding ₹ 1,50,000
Above ₹. 2,50,000	₹. 25,000+30% of the income exceeding ₹.2,50,000

Also create a main() function and create 2 objects to calculate tax for 2 employees.

Ans.

```
import java.util.*;
class Employee
```

```
{
   String pan, name;
   double tax_income,tax;
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter your PAN no.:");
      pan=sc.nextLine();
      System.out.println("Enter your name:");
      name=sc.nextLine();
      System.out.println("Enter taxable income:");
      tax_income=sc.nextDouble();
   }
   void display()
   {
      System.out.println("Pan Number\t\tName\t\tTax-income\t\tTax");
      System.out.println(pan+"\t\t"+name+"\t\t"+tax_income+"\t\t"++ tax);
   }
   void calc()
   {
      if(tax_income<=100000)
         tax=0;
      else if(tax_income>100000 && tax_income<=150000)
         tax=10/100.0*(tax_income-100000);
      else if(tax_income>150001 && tax_income<=250000)
         tax=5000+20/100.0*(tax_income-150000);
      else
         tax=25000+30/100.0*(tax_income-250000);
   }
   public static void main(String args[])
   {
      Employee ob1=new Employee ();
      Employee ob2=new Employee ();
      ob1.input();
      ob2.input();
      ob1.calc();
      ob2. calc ();
      ob1.display();
      ob2. display ();
   }
}
```

10. Define a class Salary described as below that calculates the tax depending upon the salary of a teacher:

private data Members : Name, Address, Phone, Subject Specialization, Monthly Salary, Income Tax.

public Member methods :
(i) To accept the details of a teacher including the monthly salary.
(ii) To display the details of the teacher.
(iii) To compute the annual Income Tax as 5% of the annual salary above ₹ 1,75,000/-.
Also create a main() function and create 2 objects to calculate tax for 2 teachers.

Ans.

```
import java.util.*;
class Salary
{
   String Name, Address,subSpe;
   double mSal,it;
   long phone;
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter your name:");
      Name=sc.nextLine();
      System.out.println("Enter your address:");
      Address=sc.nextLine();
      System.out.println("Enter Subject Specialization:");
      subSpe=sc.nextLine();
      System.out.println("Enter Phone No.:");
      phone=sc.nextLong();
      System.out.println("Enter monthly salary:");
      mSal=sc.nextDouble();
   }
   void display()
   {
      System.out.println("Name:"+Name);
      System.out.println("Address:"+Address);
      System.out.println("Subject Specialization:"+subSpe);
      System.out.println("Phone No.:"+ phone);
      System.out.println("Monthly salary:"+mSal);
   }
   void calc()
   {
      double aSal;
      aSal=12*mSal;
      if(aSal>175000)
         it=5/100.0*(aSal-175000);
```

```
        else
            it=0;
    }

    public static void main(String args[])
    {
        Salary ob1=new Salary();
        Salary ob2=new Salary();
        ob1.input();
        ob1.calc();
        ob1.display();
        ob2.input();
        ob2.calc();
        ob2.display();
    }
}
```

11. A special number is a number in which the sum of the factorial of each digit is equal to the number itself. For example, 145=1!+4!+5! =1+24+120

Design a class Special to check if a given number is a special number using the given members:
Class name : Special

- Class member:

n :Integer

- Static block:

Initialize n with 0

- Class functions:
  - o int factorial(int p) : calculate and return the factorial of p.
  - o void isSpecial() : check and display if the number 'n' is a special number.

Also create another class named Check that will contain the main() function to input a number and check whether the number is a Special Number or not.

Ans.

```
import java.util.*;
class Special
{
    static int n;
    static
    {
        n=0;
    }
    static int factorial(int p)
    {
        int i,f=1;
        for(i=1;i<=p;i++)
            f=f*i;
```

```
        return f;
    }
    static void isSpecial()
    {
        int i,d,s=0;
        for(i=n;i>0;i/=10)
        {
            d=i%10;
            s+=factorial(d);
        }
        if(s==n)
            System.out.println("Special Number");
        else
          System.out.println("Not a Special Number");
    }
}

public class Check
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number:");
        Special.n=sc.nextInt();
        Special.isSpecial();
    }
}
```

Chapter 8
Arrays

A. Answer the following questions.

1. What are arrays?

Ans. Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.

2. How do you declare single and 2-dimensional arrays?

Ans. General syntax of declaring a single dimensional array:

type array-name [ ] = new type [size];

General syntax of declaring a two dimensional array:

type array-name[][]=new type[row-size][column-size];

3. How do you create memory spaces for single and 2-dimensional arrays?

Ans. In a single dimensional array memory is allocated in contiguous spaces. For 2 dimensional arrays also memory are allocated in contiguous memory locations, however they are logically arranged in the form of rows and columns.

4. How do you initialise single and 2-dimensional arrays? Explain with examples.

Ans. The general form of Initialisation of single dimensional array is:

type array-name[ ]={ list of values };

For example,

int n [ ] = {5, 12, 7};

The general form of Initialisation of two dimensional array is:

type array-name[ ][ ]={ {list of values},{list of values},{list of values}... };

For example,

int a[][]={{5,1,3},{7,2,6}};

5. With the help of an example, show how arrays in Java are dynamic in nature.

Ans. Java arrays are dynamic in nature:

For example,

```
int n;
Scanner sc=new Scanner (System.in);
System.out.println("Enter the size;");
n=sc.nextInt(); //Enter the size from the user
int arr [ ]=new int[n]; //declare an array of size (dynamic)
```

As you can see the memory for the array arr is created only after taking the size 'n' from the user.

6. How do you determine the length of an array?

Ans. To determine the length of an array, the following syntax may be used:

<array-name>.length

The length is the property of an array that provides the number of elements in an array.

7. With the help of an example show how arrays are passed to a function as Call by Reference.

Ans. The following illustrates how are arrays passed as call by reference:

```
class ArrayDemo
{
        //Method to double the value of each element in the array.
        static void doubleIt(int a[])
        {
                for(int i=0; i<a.length; i++)
                {
                        a[i]=2*a[i];
                }
        }
        public static void main (String args[])
        {
                int a []={4,5,12,7,8,3};
                System.out.println("Original Array...");
                for (inti=0;i<a. length;i++)
                System.out.print(a[i]+"/t");
                doubleIt(a); // this is how you pass an array as
                //parameter.
                System.out.println ("InUpdated Array...");
                for(int i=0;i<a.length;i++)
                System.out.print(a[i]+"\t");
        }
}
```

Output when main( ) is executed will be,

Original Array....

4 5 12 7 8 3

Updated Array...

8 10 24 14 16 6

8. Explain:

i. Linear Search

ii. Binary Search

Ans.

i. Linear search or sequential search is a method for finding a particular value in a list, that consists of checking everyone of its elements, one at a time and in sequence, until the desired one is found.

ii. Binary search in Java using divide and conquer technique. A binary search or half-interval search algorithm finds the position of a specified value(the input"key") within a sorted array.

9. Compare Selection Sorting and Bubble sorting.

Ans.

Selection Sorting	Bubble Sorting
In this technique, each element is checked with the remaining elements after it.	In this technique, every element is checked with the adjacent element.
The sorting occurs from the front end of the array.	The sorting occurs from the rear end of the array.
There is only one interchange that occurs with each pass.	There may be multiple interchanges that might occur with each pass.
It is not possible at the middle of the iteration after a pass, whether the array got sorted or not.	It is possible at the middle of the iteration after a pass, whether the array got sorted or not.

10. What is the difference between one-dimensional and two-dimensional arrays?

Ans. A one-dimensional array is a list of variables with the same datatype, whereas the two-Dimensional array is 'array of arrays' having similar data types.

11. With the help of an example, show how a two-dimensional float type array of size 3x4 can be initialised.

Ans. float a[]=new float[3][4];

12. Why do you think in most cases nested loops are required to access the individual elements of a 2-dimensional array?

Ans. This is to traverse through each element of a 2D array through each row and column.

13. How do you determine the size of a two dimensional array?

Ans. No. Of elements in a 2D array is the generally the product of the no. Of rows and no. Of columns.

14. State the advantages and disadvantages of using arrays.

Ans.

Advantages

i. Gives a concise code as declaration, allocation of space and Initialisation can be done in a single line.
ii. Java arrays are dynamic, unlike other languages like C, C++.
iii. It is possible to directly access any element, provided the index is known. Thus making manipulation easy.

Disadvantages

i. Arrays allows you to store elements of only one data type, i.e., only homogenous data can be stored, thus data of heterogenous data type cannot be stored, which is often required by a programmer.
ii. Arrays always require contiguous free memory location to be allocated for storing elements. But there may be situations when the available memory is discrete in nature and therefore array cannot be used to efficiently use the memory.
iii. Careful design is required to make sure that large lists of data can be stored.

B. Answer as directed:

1. What is the difference between these two statements:

i. int sum[]=new int[10] ii. sum[1]=10;

Ans. The first statement declares an integer array of size 10 having the name 'sum'. The second statement initialises the second element of the array 'sum' with 10.

2. Write the correct form of the following arrays:

i. A1(2) ii. Name[2,5]
iii. Roll[2;] iv. Matrix(5)(5)

Ans. i. A1[2]

ii. Name[2][5]
iii. Roll[2]
iv. Matrix[5][5]

3. Write the value of individual array element.

i. int c[]={78,23,45,12,16};
ii. int p[][]={{1,1},{3,3},{5,5},{7,7}};

Ans.

i. c[0]=78, c[1]=23, c[2]=45, c[3]=12, c[4]=16
ii. p[0][0]=1, p[0][1]=1, p[1][0]=3, p[1][1]=3, p[2][0]=5, p[2][1]=5, p[3][0]=7, p[3][1]=7

4. Give the proper array declaration for the following:

i. Declare an integer array, which can hold 25 values.
ii. Declare a two dimensional array called mat 3×4 of integer.
iii. Declare and initialise a two dimensional array of size 5×5 to 0.

Ans. i. int a[]=new int[25];

ii. int mat[][]=new int[3][4];
iii. int a[][]={{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};

5. What will be the output of the following program:

```
class Output
{
    public static void main(String args[])
    {
        int a[]={6,5,4,3,2,1};
        int x;
        for(x=5;x>=0;x--)
        {
            System.out.println(a[x]);
        }
    }
}
```

Ans.

Output:

1
2
3
4
5
6

6. What will be the output of the following program:

```
class First
{
    public static void main(String args[])
    {
        int a[]={5,1,15,20,25};
        int i,j;
        int m;
        i=++a[1];
        j=a[2]++;
        m=a[i++];
        System.out.print(i+" "+j+" "+m);
    }
}
```

Ans. Output:

3 15 16

7. What will be the output of the following program, when Method invoke() is called:

```
public class StringArray
{
    void change(String arr[])
    {
    for(int i=0;i<arr.length;i++)
    {
        arr[i]=arr[i].substring(0,1).toUpperCase()+arr[i].substring(1);
    }
    }
    void invoke()
    {
        String ar[]={"kolkata","gangtok","banglore"};
        for(int i=0;i<ar.length;i++)
        System.out.println(ar[i]);
        change(ar);
        for(int i=0;i<ar.length;i++)
        System.out.println(ar[i]);
    }
}
```

Ans. Output:

```
kolkata
gangtok
banglore
Kolkata
Gangtok
Banglore
```

8. What will be the output of the following program:

```
class Output
{
    public static void main(String args[])
    {
        int a,b=0;
        int c[]={1,2,3,4,5,6,7,8,9,10};
        for(a=0;a<10;a++)
        {
            if(a%2==0)
            b+=c[a];
        }
            System.out.print(b);
    }
}
```

In what statement, state what the above program is doing?

Ans. Output:

```
25
```

9. Find the syntax error(s), if any in the following program.

```
Class First
{
    public static void main(String args[])
    {
        int sum[2,4];
        for(i=0;i<2;i++)
        {
            for(j=0;j<=3;j++)
            {
                System.print(sum);
            }
        }
    }
}
```

Ans.

```
class First
{
    public static void main(String args[])
    {
        int sum[][]=new int[2][4];
        for(i=0;i<2;i++)
        {
            for(j=0;j<=3;j++)
            {
                System.out.print(sum[i][j]+" ");
            }
        }
    }
}
```

10. Identify error(s), if any, in the following program.

```
class First
{
    public static void main(String args[])
    {
        int i;
        int a[6]={0,1,8,7,6,4};
        for(i=0;i<=a.length();i++)
        System.out.println(a[i]);
    }
}
```

Ans.

```
class First
{
    public static void main(String args[])
    {
        int i;
        int a[]={0,1,8,7,6,4};
        for(i=0;i<a.length;i++)
        System.out.println(a[i]);
    }
}
```

SECTION B

1. Write a program to input 10 numbers into an integer array and find the sum of all numbers in it.

Ans.

```
import java.util.*;
class Sol1
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,s=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();

      for(i=0;i<10;i++)
         s+=a[i];
      System.out.println("Sum="+s);
   }
}
```

2. Write a program to input 10 numbers into an integer array and find the sum of even and odd numbers separately.

Ans.

```
import java.util.*;
class Sol2
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,se=0,so=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();

      for(i=0;i<10;i++)
      {
         if(a[i]%2==0)
            se+=a[i];
         else
            so+=a[i];
      }
```

```
        System.out.println("Sum of even numbers="+se);
        System.out.println("Sum of odd numbers="+so);
    }
}
```

3. Write a program to input 10 numbers into an integer array and print those numbers which are less than its average.

Ans.

```
import java.util.*;
class Sol3
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,s=0;
        float av;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();

        for(i=0;i<10;i++)
        {
            if(a[i]%2==0)
                s+=a[i];
        }
        av=(float)s/10;
        System.out.println("Numbers less than average:");
        for(i=0;i<10;i++)
        {
            if(a[i]<av)
                System.out.print(a[i]+" ");
        }
    }
}
```

4. Write a program to input 10 numbers into an integer array and find the sum of prime numbers only.

Ans.

```
import java.util.*;
class Sol4
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,s=0,j,c;
```

```
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
          a[i]=sc.nextInt();

        for(i=0;i<10;i++)
        {
          c=0;
          for(j=1;j<=a[i];j++)
          {
            if(a[i]%j==0)
              c++;
          }
          if(c==2)
            s+=a[i];
        }
          if(s>0)
          System.out.println("Sum of prime numbers:"+s);
          else
          System.out.println("No prime numbers found");
    }
}
```

5. Write a program to input 10 numbers into an integer array and check whether all numbers are 3-digit numbers or not.

Ans.

```
import java.util.*;
class Sol5
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[]=new int[10],i;
    boolean f=true;
    System.out.println("Enter 10 numbers:");
    for(i=0;i<10;i++)
    a[i]=sc.nextInt();

    for(i=0;i<10;i++)
    {
      if(!(a[i]>=100 && a[i]<=999))
      f=false;
    }
    if(f)
    System.out.println("All are 3 digit numbers");
```

```
        else
        System.out.println("All are not 3 digit numbers");
    }
}
```

6. Write a program to input 10 numbers into an integer array and check whether all numbers in it are same or not.

Ans.

```
import java.util.*;
class Sol6
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i;
        boolean f=true;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();

        for(i=0;i<10;i++)
        {
          if(a[i]!=a[0])
            f=false;
        }
        if(f)
            System.out.println("All are same");
        else
            System.out.println("All are not same");
    }
}
```

7. Write a program to input 10 numbers into an integer array and check whether they are in ascending order or not.

Ans.

```
import java.util.*;
class Sol7
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i;
        boolean f=true;
```

```
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
           a[i]=sc.nextInt();

        for(i=0;i<9;i++)
        {
          if(a[i]>a[i+1])
           f=false;
        }
        if(f)
           System.out.println("All are in ascending order");
        else
           System.out.println("All are not in ascending order");
    }
}
```

8. Write a program to input 10 numbers into an integer array and find the position of the largest and the smallest number.

Ans.

```
import java.util.*;
class Sol8
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,l=0,s=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      l=s=a[0];
      for(i=1;i<10;i++)
      {
        if(a[i]>l)
         l=a[i];
        if(a[i]<s)
         s=a[i];
      }
      for(i=0;i<10;i++)
      {
         if(a[i]==l)
            System.out.println("Position of largest number:"+i);
```

```
            if(a[i]==s)
               System.out.println("Position of smallest number:"+i);
         }
      }
   }
```

9. Write a program to input 10 numbers into an integer array and interchange the largest number with the smallest number within the array and print the modified array. Assume that there is only one largest and smallest number.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	67	34	15	16	89	15

After interchange it should have the elements arranged as:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	89	7	67	34	15	16	3	15

Ans.

```
import java.util.*;
class Sol9
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,l=0,s=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      l=s=a[0];
      for(i=1;i<10;i++)
      {
        if(a[i]>l)
         l=a[i];
        if(a[i]<s)
         s=a[i];
      }
      for(i=0;i<10;i++)
      {
         if(a[i]==l)
            a[i]=s;
         else if(a[i]==s)
            a[i]=l;
      }
```

```
        System.out.println("Modified Array:");
        for(i=0;i<10;i++)
        {
           System.out.print(a[i]+" ");
        }
     }
}
```

10. Write a program to input 10 numbers into an integer array and replace all prime numbers in it (if any) by 0 and print the modified array.

For example, if the array contains the following elements:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	67	34	15	16	89	13

After replacing the prime numbers with 0 the modified array should have the elements as:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	0	0	67	34	15	16	89	0

Ans.

```
import java.util.*;
class Sol10
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,j,c;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();

      for(i=0;i<10;i++)
      {
        c=0;
        for(j=1;j<=a[i];j++)
        {
           if(a[i]%j==0)
            c++;
        }
        if(c==2)
         a[i]=0;
      }

      System.out.println("Modified Array:");
      for(i=0;i<10;i++)
```

```
        {
            System.out.print(a[i]+" ");
        }
    }
}
```

11. Write a program to input 10 numbers into an integer array and print the smallest prime number in the array.

```
import java.util.*;
class Sol11
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,j,c,s=0;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();

        for(i=0;i<10;i++)
        {
          c=0;
          for(j=1;j<=a[i];j++)
          {
             if(a[i]%j==0)
              c++;
          }
          if(c==2)
          {
             if(s==0)
              s=a[i];
             if(a[i]<s)
              s=a[i];
          }
        }
        if(s>0)
            System.out.println("Smallest prime number:"+s);
        else
            System.out.println("No prime number found");
    }
}
```

12. Write a program to input 10 numbers into an integer array and print the position of the second largest number in it.

```
import java.util.*;
class Sol12
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,l=0,sl=0,p=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      l=a[0];
      for(i=1;i<10;i++)
      {
        if(a[i]>l)
         l=a[i];
      }

      for(i=0;i<10;i++)
      {
        if(a[i]!=l)
        {
            if(sl==0)
            {
               sl=a[i];
               p=i;
            }
            if(a[i]>sl)
            {
               sl=a[i];
               p=i;
            }
         }
      }
      System.out.println("Position of second largest:"+p);
   }
}
```

13. Write a program to input 10 numbers into an integer array and arrange the numbers in descending order using Linear Sorting technique.

```
import java.util.*;
class Sol13
{
   static void main()
```

```
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,j,t;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();
        for(i=0;i<9;i++)
        {
            for(j=i+1;j<10;j++)
            {
                if(a[i]<a[j])
                {
                    t=a[i];
                    a[i]=a[j];
                    a[j]=t;
                }
            }
        }
        System.out.println("Sorted Array:");
        for(i=0;i<10;i++)
        {
            System.out.print(a[i]+" ");
        }
    }
}
```

14. Write a program to input 10 numbers into a float type array and arrange the numbers in descending order using Bubble Sorting technique.

```
import java.util.*;
class Sol14
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,j,t;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();
        for(i=9;i>0;i--)
        {
            for(j=0;j<i;j++)
            {
                if(a[j]<a[j+1])
```

```
            {
               t=a[j];
               a[j]=a[j+1];
               a[j+1]=t;
            }
         }
      }
      System.out.println("Sorted Array:");
      for(i=0;i<10;i++)
      {
         System.out.print(a[i]+" ");
      }
   }
}
```

15. Write a program to input 10 numbers into an integer array and reverse the array and print the modified array.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	89	34	15	16	67	25

After reversal it should have the elements arranged as:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
25	67	16	15	34	89	7	3	12	9

Ans.

```
import java.util.*;
class Sol15
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,t;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<5;i++)
      {
         t=a[i];
         a[i]=a[9-i];
         a[9-i]=t;
      }
```

```
        System.out.println("Modified Array:");
        for(i=0;i<10;i++)
        {
           System.out.print(a[i]+" ");
        }
    }
}
```

16. Write a program to input 10 numbers into an integer array and interchange the consecutive elements in it. That is, interchange a[0] with a[1], a[2] with a[3], a[4] with a[5] …

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	89	34	15	16	67	25

After interchange it should have the elements arranged as:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
12	9	7	3	34	89	16	15	25	67

Ans.

```
import java.util.*;
class Sol16
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,t;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<9;i+=2)
      {
         t=a[i];
         a[i]=a[i+1];
         a[i+1]=t;
      }
      System.out.println("Modified Array:");
      for(i=0;i<10;i++)
      {
         System.out.print(a[i]+" ");
      }
   }
}
```

17. Write a program to input 10 numbers into an integer array and find the frequency of the largest number.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	89	3	7	89	34	15	16	67	89

Output should be:

Frequency of largest number = 3

Ans.

```
import java.util.*;
class Sol17
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,l=0,c=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<9;i++)
      {
         if(a[i]>l)
            l=a[i];
      }
      for(i=0;i<10;i++)
      {
         if(a[i]==l)
            c++;
      }
      System.out.print("Frequency of the largest:"+c);
   }
}
```

18. Write a program to input 10 numbers into an integer array and input a position. Now delete the element at that position by shifting the rest of the numbers to the left and insert a 0 at the end.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	89	34	15	16	67	25

And element at position 4 is to be deleted, the resultant array should be:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	34	15	16	67	25	0

```
import java.util.*;
class Sol18
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,p;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      System.out.println("Enter the position to delete:");
      p=sc.nextInt();

      for(i=p;i<9;i++)
      {
         a[i]=a[i+1];
      }
      a[9]=0;
      System.out.println("Modified Array:");
      for(i=0;i<10;i++)
      {
         System.out.print(a[i]+" ");
      }
   }
}
```

19. Write a program to input 10 numbers into an integer array and input a number and a position. Now insert the number at that position by shifting the rest of the numbers to the right. The last element is therefore removed from the array.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	89	34	15	16	67	25

And if 36 is to be inserted at position 4, the resultant array should be:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	36	89	34	15	16	67

Notice that the last element i.e. 25 got removed after the shifting process.

Ans.

```
import java.util.*;
class Sol19
{
   static void main()
   {
```

```
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,p,n;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
           a[i]=sc.nextInt();

        System.out.println("Enter the number to insert:");
        n=sc.nextInt();

        System.out.println("Enter the position to insert:");
        p=sc.nextInt();

        for(i=9;i>p;i--)
        {
           a[i]=a[i-11];
        }
        a[p]=n;
        System.out.println("Modified Array:");
        for(i=0;i<10;i++)
        {
           System.out.print(a[i]+" ");
        }
     }
  }
```

20. Write a program to input 10 positive or negative numbers (no zero) into an integer array and shift all positive numbers to the beginning of the array and negative numbers to the end of the array; without changing the order of the numbers.

For example, if the array contains the following elements:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
-9	12	-3	-7	89	-34	15	16	-67	25

After shifting the array should contain the elements arranged in the following manner:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
12	89	15	16	25	-9	-3	-7	-34	-67

Positive numbers Negative numbers

Ans.

```
import java.util.*;
class Sol20
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
```

```
        int a[]=new int[10],i,j,t;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
          a[i]=sc.nextInt();
        for(i=9;i>0;i--)
        {
          for(j=0;j<i;j++)
          {
            if(a[j]<0 && a[j+1]>0)
            {
              t=a[j];
              a[j]=a[j+1];
              a[j+1]=t;
            }
          }
        }
        System.out.println("Modified Array:");
        for(i=0;i<10;i++)
        {
          System.out.print(a[i]+" ");
        }
      }
    }
```

21. Write a program to input 10 numbers into an integer array and shift all even numbers to the beginning of the array and odd numbers to the end of the array; without changing the order of the numbers.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	89	34	15	16	67	24

After shifting the resultant array should be:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
12	34	16	24	9	3	7	89	15	67

Ans.

```
import java.util.*;
class Sol21
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
```

```
        int a[]=new int[10],i,j,t;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
          a[i]=sc.nextInt();
        for(i=9;i>0;i--)
        {
          for(j=0;j<i;j++)
          {
            if(a[j]%2!=0 && a[j+1]%2==0)
            {
              t=a[j];
              a[j]=a[j+1];
              a[j+1]=t;
            }
          }
        }
        System.out.println("Modified Array:");
        for(i=0;i<10;i++)
        {
          System.out.print(a[i]+" ");
        }
    }
}
```

22. Write a program to input 10 numbers into an integer array and print those single-digit positive numbers which are not present in the array.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
9	12	3	7	89	34	9	16	8	24

Output should be:
1, 2, 4, 5, 6

Ans.

```
import java.util.*;
class Sol22
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[]=new int[10],i,j;
    boolean f;
    System.out.println("Enter 10 numbers:");
```

```
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();
        for(i=0;i<=9;i++)
        {
            f=false;
            for(j=0;j<10;j++)
            {
                if(a[j]==i)
                    f=true;
            }
            if(f==false)
                System.out.println(i);
        }
    }
}
```

23. Write a program to input 10 numbers into an integer array and find the frequency of each two-digit numbers present in it.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
99	12	34	7	99	34	12	16	12	24

Output should be:

Frequency of 12 = 3
Frequency of 16 = 1
Frequency of 24 = 1
Frequency of 34 = 2
Frequency of 99 = 2

Ans.

```
import java.util.*;
class Sol23
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,j,f;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();
        for(i=10;i<=99;i++)
        {
            f=0;
            for(j=0;j<10;j++)
            {
```

```
            if(a[j]==i)
               f++;
         }
         if(f>0)
            System.out.println("Frequency of "+i+" is="+f);
      }
   }
}
```

24. Write a program to input 10 numbers into an integer array and find the frequency of each number present in it.

Ans.

```
import java.util.*;
class Sol24
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,j,f,c;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<10;i++)
      {
         c=f=0;
         for(j=0;j<10;j++)
         {
            if(a[i]==a[j])
            {
               c++;
               if(j<i)
                  f=1;
            }
         }
         if(f==0)
            System.out.println("Frequency of "+a[i]+" is="+c);
      }
   }
}
```

25. Write a program to input 10 numbers into an integer array and print the number having maximum frequency assume that there is only one number having maximum frequency.

For example, if array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
99	12	34	7	99	34	12	16	12	24

Output should be:
Number having maximum frequency=12

Ans.

```
import java.util.*;
class Sol25
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,j,max=0,n=0,c;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<10;i++)
      {
         c=0;
         for(j=0;j<10;j++)
         {
            if(a[i]==a[j])
               c++;
         }
         if(c>max)
         {
            max=c;
            n=a[i];
         }
      }
      System.out.println("Number having maximum frequency="+n);
   }
}
```

26. Write a program to input 10 numbers into an integer array and store only the even numbers into another array and display it.

For example, if the given array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
99	12	34	7	99	36	12	16	14	123

The resultant array should be:

b[0]	b[1]	b[2]	b[3]	b[4]	b[5]
12	34	36	12	16	14

Ans.

```
import java.util.*;
class Sol26
```

```
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,b[]=new int[10],c=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<10;i++)
      {
         if(a[i]%2==0)
            b[c++]=a[i];
      }
      System.out.println("Resultant Array:");
      for(i=0;i<c;i++)
      {
         System.out.print(b[i]+ " ");
      }
   }
}
```

27. Write a program to input 10 numbers into an integer array and store only the unique numbers into another array and display it.

For example, if the given array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
99	12	34	7	99	36	12	16	14	123

The resultant array should be:

b[0]	b[1]	b[2]	b[3]	b[4]	b[5]
34	7	36	16	14	123

Ans.

```
import java.util.*;
class Sol27
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,b[]=new int[10],c=0,j,p=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<10;i++)
      {
         p=0;
```

```
            for(j=0;j<10;j++)
            {
               if(a[i]==a[j])
                  p++;
            }
            if(p==1)
               b[c++]=a[i];
         }
         System.out.println("Resultant Array:");
         for(i=0;i<c;i++)
         {
            System.out.print(b[i]+ " ");
         }
      }
   }
```

28. Write a program to input 10 numbers into an integer array and store each number only once into another array irrespective of the number of times it is present in the array.

For example, if the given array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
99	12	34	7	99	36	12	16	14	16

The resultant array should be:

b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]
99	12	34	7	36	16	14

```
import java.util.*;
class Sol28
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10],i,b[]=new int[10],c=0,j,p=0;
      System.out.println("Enter 10 numbers:");
      for(i=0;i<10;i++)
         a[i]=sc.nextInt();
      for(i=0;i<10;i++)
      {
         p=0;
         for(j=i+1;j<10;j++)
         {
            if(a[i]==a[j])
               p++;
         }
         if(p==0)
```

```
            b[c++]=a[i];
        }
        System.out.println("Resultant Array:");
        for(i=0;i<c;i++)
        {
            System.out.print(b[i]+" ");
        }
    }
}
```

29. Write a program to input 10 numbers into an integer array and store all even numbers into one array and all odd numbers into another array. Display all the three arrays.

For example, if the given array contains

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
99	12	34	7	87	36	94	16	14	45

The resultant arrays should be:

b[0]	b[1]	b[2]	b[3]	b[4]	b[5]
12	34	36	94	16	14

c[0]	c[1]	c[2]	c[3]
99	7	87	45

Ans.

```
import java.util.*;
class Sol29
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,b[]=new int[10],c[]=new int[10];
        int ib=0,ic=0;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();
        for(i=0;i<10;i++)
        {
            if(a[i]%2==0)
                b[ib++]=a[i];
            else
                c[ic++]=a[i];
        }
        System.out.println("Resultant Array Even No.s:");
```

```
        for(i=0;i<ib;i++)
        {
           System.out.print(b[i]+" ");
        }
        System.out.println("Resultant Array Odd No.s:");
        for(i=0;i<ic;i++)
        {
           System.out.print(c[i]+" ");
        }
     }
  }
```

30. Write a program to input numbers into a 5×5 integer matrix and find the sum of all numbers in it.

Ans.

```
import java.util.*;
 class Sol30
 {
    static void main()
    {
       Scanner sc=new Scanner(System.in);
       int a[][]=new int[5][5],i,j,s=0;
       System.out.println("Enter 25 numbers:");
       for(i=0;i<5;i++)
       {
          for(j=0;j<5;j++)
          {
             a[i][j]=sc.nextInt();
          }
       }
       for(i=0;i<5;i++)
       {
          for(j=0;j<5;j++)
          {
             s+=a[i][j];
          }
       }
       System.out.print("Sum="+s);
    }
 }
```

31. Write a program to input numbers into a 5×5 integer matrix and print the largest and smallest number from it.

Ans.

```
import java.util.*;
 class Sol31
 {
```

```
    static void main()
    {
       Scanner sc=new Scanner(System.in);
       int a[][]=new int[5][5],i,j,l=0,s=0;
       System.out.println("Enter 25 numbers:");
       for(i=0;i<5;i++)
       {
          for(j=0;j<5;j++)
          {
             a[i][j]=sc.nextInt();
          }
       }
       for(i=0;i<5;i++)
       {
          for(j=0;j<5;j++)
          {
             if(i==0 && j==0)
                l=s=a[i][j];
             else
             {
                if(a[i][j]>l)
                   l=a[i][j];
                if(a[i][j]<s)
                   s=a[i][j];
             }
          }
       }
       System.out.print("Largest="+l);
       System.out.print("Smallest="+s);
    }
}
```

32. Write a program to input numbers into a 5×5 integer matrix and interchange the largest number with the smallest number and display the modified matrix.

For example. if the given matrix is:

5	12	13	45	22
17	67	48	19	2
100	17	10	128	79
11	28	156	59	63
72	29	37	59	71

The resultant matrix should be:

5	12	13	45	22
17	67	48	19	156
100	17	10	128	79
11	28	2	59	63
72	29	37	59	71

Ans.

```
import java.util.*;
class Sol32
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[5][5],i,j,l=0,s=0;
      System.out.println("Enter 25 numbers:");
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            a[i][j]=sc.nextInt();
         }
      }
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            if(i==0 && j==0)
               l=s=a[i][j];
            else
            {
               if(a[i][j]>l)
                  l=a[i][j];
               if(a[i][j]<s)
                  s=a[i][j];
            }
         }
      }
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
              if(a[i][j]==l)
                  a[i][j]=s;
```

```
                else if(a[i][j]==s)
                    a[i][j]=l;

            }
        }
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                System.out.print("\t"+a[i][j]);
            }
            System.out.println();
        }
    }
}
```

33. Write a program to input numbers into a 5×5 integer matrix and find the sum of each row separately.

For example, if the given matrix is:

1	2	3	5	2
7	6	8	9	2
1	7	11	3	4
3	2	5	9	8
5	5	3	2	1

Output:
Sum of row 1 = 13
Sum of row 2 = 32
Sum of row 3 = 26
Sum of row 4 = 27
Sum of row 5 = 16

Ans.

```
import java.util.*;
class Sol33
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[5][5],i,j,s=0;
      System.out.println("Enter 25 numbers:");
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
```

```
            a[i][j]=sc.nextInt();
        }
    }
    for(i=0;i<5;i++)
    {
        s=0;
        for(j=0;j<5;j++)
        {
            s+=a[i][j];
        }
        System.out.println("Sum of row"+(i+1)+ "=" +s);
    }
  }
}
```

34. Write a program to input numbers into a 5×5 integer matrix and find the sum of each column separately.

For example, if the given matrix is:

1	2	3	5	2
7	6	8	9	2
1	7	11	3	4
3	2	5	9	8
5	5	3	2	1

Output:
Sum of column 1 = 17
Sum of column 2 = 22
Sum of column 3 = 30
Sum of column 4 = 28
Sum of column 5 = 17

Ans.

```
import java.util.*;
class Sol34
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[][]=new int[5][5],i,j,s=0;
    System.out.println("Enter 25 numbers:");
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
      {
        a[i][j]=sc.nextInt();
```

```
        }
      }
      for(i=0;i<5;i++)
      {
        s=0;
        for(j=0;j<5;j++)
        {
          s+=a[j][i];
        }
        System.out.println("Sum of column" +(i+1)+ "= "+s);
      }
    }
}
```

35. Write a program to input numbers into a 5×5 integer matrix and check whether all numbers in it are even numbers or not.

Example 1:

If the given matrix is:

6	12	14	46	22
18	66	48	18	2
100	16	10	128	80
8	28	156	60	64
72	30	38	58	72

Output:

All are even numbers.

Example 2:

If the given matrix is:

6	12	14	45	22
18	68	48	22	146
100	27	10	128	78
49	81	35	60	63
72	30	40	70	78

Output:

All are not even numbers.

Ans.

```
import java.util.*;
class Sol35
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
```

```
        int a[][]=new int[5][5],i,j,f=0;
        System.out.println("Enter 25 numbers:");
        for(i=0;i<5;i++)
        {
          for(j=0;j<5;j++)
          {
            a[i][j]=sc.nextInt();
          }
        }
        for(i=0;i<5;i++)
        {
          for(j=0;j<5;j++)
          {
            if(a[i][j]%2!=0)
              f=1;
          }
        }
        if(f==0)
          System.out.println("All are even numbers");
        else
          System.out.println("All are not even numbers");
    }
}
```

36. Write a program to input numbers into a 5×5 integer matrix and print the largest and the smallest number among both the diagonals.

Ans.

```
import java.util.*;
class Sol36
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[][]=new int[5][5],i,j,l=0,s=0;
    System.out.println("Enter 25 numbers:");
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
      {
        a[i][j]=sc.nextInt();
      }
    }
    l=s=a[0][0];
    for(i=0;i<5;i++)
    {
```

```
            if(a[i][i]>l)
                l=a[i][i];
            if(a[i][i]<s)
                s=a[i][i];
            if(a[i][4-i]>l)
                l=a[i][4-i];
            if(a[i][4-i]<s)
                s=a[i][4-i];
        }
        System.out.println("Largest="+l);
        System.out.println("Smallest="+s);
    }
}
```

37. Write a program to input numbers into a 5×5 integer matrix and sort the major diagonal elements in descending order.

Ans.

```
import java.util.*;
class Sol37
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[][]=new int[5][5],i,j,t;
        System.out.println("Enter 25 numbers:");
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                a[i][j]=sc.nextInt();
            }
        }
        for(i=0;i<5;i++)
        {
            for(j=i+1;j<5;j++)
            {
                if(a[i][i]<a[j][j])
                {
                    t=a[i][i];
                    a[i][i]=a[j][j];
                    a[j][j]=t;
                }
            }
        }
        System.out.println("Sorted Array:");
```

```
        for(i=0;i<5;i++)
        {
          for(j=0;j<5;j++)
          {
              System.out.print("\t"+a[i][j]);
          }
          System.out.println();
        }
    }
}
```

38. Write a program to input numbers into a 5×5 integer matrix and sort the minor diagonal elements in ascending order.

Ans.

```
import java.util.*;
class Sol38
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[5][5],i,j,t;
      System.out.println("Enter 25 numbers:");
      for(i=0;i<5;i++)
      {
        for(j=0;j<5;j++)
        {
          a[i][j]=sc.nextInt();
        }
      }
      for(i=0;i<5;i++)
      {
        for(j=i+1;j<5;j++)
        {
          if(a[i][4-i]<a[j][4-j])
          {
            t=a[i][4-i];
            a[i][4-i]=a[j][4-j];
            a[j][4-j]=t;
          }
        }
      }
      System.out.println("Sorted Array:");
      for(i=0;i<5;i++)
      {
        for(j=0;j<5;j++)
```

```
        {
            System.out.print("\t"+a[i][j]);
        }
        System.out.println();
    }
  }
}
```

39. Write a program to input numbers into a 5×5 integer matrix and sort elements of each row in ascending order.

Ans.

```
import java.util.*;
class Sol39
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[][]=new int[5][5],i,j,t,r;
    System.out.println("Enter 25 numbers:");
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
      {
        a[i][j]=sc.nextInt();
      }
    }
    for(r=0;r<5;r++)
    {
      for(i=0;i<5;i++)
      {
        for(j=i+1;j<5;j++)
        {
          if(a[r][i]>a[r][j])
          {
            t=a[r][i];
            a[r][i]=a[r][j];
            a[r][j]=t;
          }
        }
      }
    }
    System.out.println("Sorted Array:");
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
```

```
        {
            System.out.print("\t"+a[i][j]);
        }
        System.out.println();
    }
  }
}
```

40. Write a program to input numbers into a 5×5 integer matrix and sort elements of each column in descending order.

Ans.

```
import java.util.*;
class Sol40
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[][]=new int[5][5],i,j,t,c;
    System.out.println("Enter 25 numbers:");
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
      {
        a[i][j]=sc.nextInt();
      }
    }
    for(c=0;c<5;c++)
    {
      for(i=0;i<5;i++)
      {
        for(j=i+1;j<5;j++)
        {
          if(a[i][c]<a[j][c])
          {
            t=a[i][c];
            a[i][c]=a[j][c];
            a[j][c]=t;
          }
        }
      }
    }
    System.out.println("Sorted Array:");
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
```

```
        {
            System.out.print("\t"+a[i][j]);
        }
        System.out.println();
    }
  }
}
```

41. Write a program to input numbers into a 5×5 integer matrix and print the number having maximum frequency.

Ans.

```
import java.util.*;
class Sol41
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[5][5],i,j,t,r,c,p,max=0,n=0;
      System.out.println("Enter 25 numbers:");
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            a[i][j]=sc.nextInt();
         }
      }
      for(r=0;r<5;r++)
      {
         for(c=0;c<5;c++)
         {
            p=0;
            for(i=0;i<5;i++)
            {
               for(j=0;j<5;j++)
               {
                  if(a[r][c]==a[i][j])
                     p++;
               }
            }
            if(p>max)
            {
               max=p;
               n=a[r][c];
            }
```

```
            }
        }
        System.out.println("No. having maximum frquency=:"+n);

    }
}
```

42. Write a program to input numbers into a 5×5 integer matrix and print the frequency of each number.

Ans.

```
import java.util.*;
class Sol42
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[][]=new int[5][5],i,j,l=0,s=0,x,c;
        System.out.println("Enter 25 numbers:");
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                a[i][j]=sc.nextInt();
            }
        }
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                if(i==0 && j==0)
                    l=s=a[i][j];
                else
                {
                    if(a[i][j]>l)
                        l=a[i][j];
                    if(a[i][j]<s)
                        s=a[i][j];
                }
            }
        }
        for(x=s;x<=l;x++)
        {
            c=0;
            for(i=0;i<5;i++)
```

```
        {
            for(j=0;j<5;j++)
            {
                if(x==a[i][j])
                    c++;
            }
        }
        if(c>0)
            System.out.println("Frequency of"+x+"is="+c);
    }
  }
}
```

43. Write a program to read in a 5×5 matrix and then ask for an input of a number and search its position in the matrix. If found, print the indices where it is found, otherwise print "Not Found".

Ans.

```
import java.util.*;
class Sol43
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[][]=new int[5][5],i,j,f=0,n;
    System.out.println("Enter 25 numbers:");
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
      {
        a[i][j]=sc.nextInt();
      }
    }
    System.out.println("Enter the number to search:");
    n=sc.nextInt();
    for(i=0;i<5;i++)
    {
      for(j=0;j<5;j++)
      {
        if(a[i][j]==n)
        {
          System.out.println("Position present at i="+i+ "j="+j);
          f=1;
        }
      }
    }
```

```
            if(f==0)
            System.out.println("Not Found");
    }
}
```

44. Write a program to input numbers into a 5×5 integer matrix and check whether it is a unit matrix or not. Unit matrix is such a matrix whose major diagonal elements are 1 and the rest are 0.

Ans.

```
import java.util.*;
class Sol44
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[][]=new int[5][5],i,j,f=0;
        System.out.println("Enter 25 numbers:");
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                a[i][j]=sc.nextInt();
            }
        }
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                if(i==j)
                {
                  if(a[i][j]!=1)
                      f=1;
                }
                else
                {
                   if(a[i][j]!=0)
                      f=1;
                }
            }
        }
        if(f==0)
            System.out.println("Unit Matrix");
        else
            System.out.println("Not a Unit Matrix");
    }
}
```

45. Write a program to input numbers into a 5×5 integer matrix and check whether it is a diagonal matrix or not. Diagonal matrix is such a matrix whose major diagonal elements are non-zeroes and rest of the elements are zeroes.

Ans.

```
import java.util.*;
class Sol45
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[5][5],i,j,f=0;
      System.out.println("Enter 25 numbers:");
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            a[i][j]=sc.nextInt();
         }
      }
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            if(i==j)
            {
              if(a[i][j]==0)
                 f=1;
            }
            else
            {
               if(a[i][j]!=0)
                  f=1;
            }
         }
      }
      if(f==0)
         System.out.println("Diagonal Matrix");
      else
         System.out.println("Not a Diagonal Matrix");
   }
}
```

46. Write a program to input numbers into a 5×5 integer matrix and find it's transpose. The transpose of a matrix is a new matrix whose rows are the columns of the original. (This makes the columns of the new matrix the rows of the original). Here is a matrix and its transpose:

The superscript "T" means transpose.

Ans.

```
import java.util.*;
class Sol46
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[5][5],i,j;
      int T[][]=new int[5][5];
      System.out.println("Enter 25 numbers:");
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            a[i][j]=sc.nextInt();
         }
      }
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            T[i][j]=a[j][i];
         }
      }
      System.out.println("Transpose Matrix:");
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            System.out.print(T[i][j]+"\t");
         }
         System.out.println();
      }
   }
}
```

47. Write a program to input numbers into a 5×5 integer matrix and check whether it is a symmetric matrix or not. Symmetric matrix is such a square matrix whose row elements are exactly same as column elements. Thus a symmetric matrix is a square matrix that is equal to its transpose.

Ans.

```
import java.util.*;
class Sol47
{
   static void main()
```

```
    {
        Scanner sc=new Scanner(System.in);
        int a[][]=new int[5][5],i,j,f=0;
        System.out.println("Enter 25 numbers:");
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                a[i][j]=sc.nextInt();
            }
        }
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                if(a[i][j]!=a[j][i])
                    f=1;
            }
        }
        if(f==0)
            System.out.println("Symmetric Matrix");
        else
            System.out.println("Not a Symmetric Matrix");
    }
}
```

48. Write a program to input numbers into a 5×5 integer matrix and check whether it is a Lower Triangular Matrix or not. Lower Triangular Matrix is such a matrix whose elements above the major diagonal are zeroes and below the major diagonal are non-zeroes.

Ans.

```
import java.util.*;
class Sol48
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[][]=new int[5][5],i,j,f=0;
        System.out.println("Enter 25 numbers:");
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                a[i][j]=sc.nextInt();
            }
        }
```

```
        for(i=0;i<5;i++)
        {
           for(j=0;j<5;j++)
           {
              if(j>i)
              {
                if(a[i][j]!=0)
                    f=1;
              }
              else if(j<i)
              {
                 if(a[i][j]==0)
                    f=1;
              }
           }
        }
        if(f==0)
           System.out.println(" Lower Triangular  Matrix");
        else
           System.out.println("Not a  Lower Triangular  Matrix");
     }
  }
```

49. Write a program to input numbers into two - 3×3 integer matrices and find their sum and store it in the third matrix.

Ans.

```
import java.util.*;
class Sol49
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[3][3],b[][]=new int[3][3],c[][]=new int[3][3],i,j;
      System.out.println("Enter the first matrix:");
      for(i=0;i<3;i++)
      {
         for(j=0;j<3;j++)
         {
            a[i][j]=sc.nextInt();
         }
      }
      System.out.println("Enter the second matrix:");
      for(i=0;i<3;i++)
      {
         for(j=0;j<3;j++)
```

```
            {
                b[i][j]=sc.nextInt();
            }
        }
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                c[i][j]=a[i][j]+b[i][j];
            }
        }
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                System.out.print(c[i][j]+"\t");
            }
            System.out.println();
        }
    }
}
```

50. Write a program to input numbers into a 5×5 integer matrix and check whether it is zero or null matrix or not. A null matrix is a matrix, whose all elements are zero.

Ans.

```
import java.util.*;
class Sol50

{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[][]=new int[5][5],i,j,f=0;
        System.out.println("Enter 25 numbers:");
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                a[i][j]=sc.nextInt();
            }
        }
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
```

```
            if(a[i][j]!=0)
               f=1;
         }
      }
      if(f==0)
         System.out.println("Null Matrix");
      else
         System.out.println("Not a  Null Matrix");
   }
}
```

51. Write a program to input numbers into a 5×5 integer matrix and check whether it is a scalar matrix or not. Scalar matrix is such a matrix whose major diagonal elements are all same.

Ans.

```
import java.util.*;
class Sol51

{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int a[][]=new int[5][5],i,j,f=0;
      System.out.println("Enter 25 numbers:");
      for(i=0;i<5;i++)
      {
         for(j=0;j<5;j++)
         {
            a[i][j]=sc.nextInt();
         }
      }
      for(i=0;i<5;i++)
      {
         if(a[i][i]!=a[0][0])
               f=1;
      }
      if(f==0)
         System.out.println("Scalar Matrix");
      else
         System.out.println("Not a  Scalar Matrix");
   }
}
```

52. Create a class named Student containing the following instance variables:

Instance Variables:

roll[ ] of int type array reference.

name[ ] of String type array reference.

Member functions:

i. Parameterized constructor to initialize the data members.

ii. To accept a roll as parameter and search for it in roll[]. If found, the corresponding name should be displayed otherwise a relevant message should be displayed.

Ans.

```
import java.util.*;
class Student
{
   int roll[];
   String name[];
   Student(int r[],String n[])
   {
      roll=r;
      name=n;
   }
   void search(int r)
   {
      int i,f=0;
      for(i=0;i<roll.length;i++)
      {
         if (roll[i]==r)
         {
            System.out.println("Name:"+name[i]);
            f=1;
         }
      }
      if(f==0)
         System.out.println("Not found");
   }
}
```

53. Create a method which accepts an integer as parameter and return true if it is a prime number, otherwise return false. Now use it in another method which accepts a two-dimensional integer array as parameter and check whether all the elements of the minor diagonal are prime numbers or not.

Ans.

```
import java.util.*;
class Sol53
{
   static boolean isPrime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
```

```
        {
            if(n%i==0)
                c++;
        }
        if(c==2)
            return true;
        else
            return false;
    }
    static void main(int a[][])
    {
        int i,j,f=0;
        for(i=0;i<a.length;i++)
        {
            for(j=0;j<a[0].length;j++)
            {
                if(i+j==a[0].length-1)
                {
                    if(isPrime(a[i][j])==false)
                        f=1;
                }
            }
        }
        if(f==0)
            System.out.println("All no.s of the minor diagonal are prime");
        else
            System.out.println("All no.s of the minor diagonal are not  prime");
    }
}
```

54. Write a program to input and sort the weight of ten people. Sort and display them in descending order using the selection sort technique.

Ans.

```
import java.util.*;
class Sol54
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a[]=new int[10],i,j,s,p;
        System.out.println("Enter 10 numbers:");
        for(i=0;i<10;i++)
            a[i]=sc.nextInt();
        for(i=0;i<9;i++)
        {
```

```
            s=a[i];p=i;
            for(j=i+1;j<10;j++)
            {
               if(a[j]<s)
               {
                  s=a[j];
                  p=j;
               }
            }
            a[p]=a[i];
            a[i]=s;
         }
         System.out.println("Sorted Array:");
         for(i=0;i<10;i++)
         {
            System.out.print(a[i]+" ");
         }
      }
   }
```

55. The annual examination results of 50 students in a class is tabulated as follows.

Write a program to read the data, calculate and display the following:

a. Average mark obtained by each student.

b. Print the roll number and average marks of the students whose average mark is above 80.

c. Print the roll number and average marks of the students whose average mark is below 40.

Ans.

```
import java.util.*;
class Sol55
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int roll[]=new int[50],i;
      int m1[]=new int[50],m2[]=new int[50],m3[]=new int[50];
      float av[]=new float[50];
      System.out.println("Enter the roll and marks:");
      for(i=0;i<50;i++)
      {
         roll[i]=sc.nextInt();
         m1[i]=sc.nextInt();
         m2[i]=sc.nextInt();
```

```
            m3[i]=sc.nextInt();
            av[i]=(float)(m1[i]+m2[i]+m3[i])/3;
        }
        System.out.println("Average marks obtained by each student:");
        System.out.println("Roll\t\tAverage");
        for(i=0;i<50;i++)
        {
            System.out.println(roll[i]+"\t\t"+av[i]);
        }
        System.out.println("Average marks above 80 obtained by each student:");
        System.out.println("Roll\t\tAverage");
        for(i=0;i<50;i++)
        {
            if(av[i]>80)
                System.out.println(roll[i]+"\t\t"+av[i]);
        }
        System.out.println("Average marks below 40 obtained by each student:");
        System.out.println("Roll\t\tAverage");
        for(i=0;i<50;i++)
        {
            if(av[i]<40)
                System.out.println(roll[i]+"\t\t"+av[i]);
        }
    }
}
```

56. Write a program to store 6 element in an array P, and 4 elements in an array Q and produce a third array R, containing all elements of array P and Q. Display the resultant array.

Ans.

```
class Sol56
{
    static void main()
    {
        int P[]={4,12,23,34,45,56};
        int Q[]={17,15,12,23};
        int R[]=new int[10],i,c=0;
        for(i=0;i<6;i++)
        {
            R[c++]=P[i];
        }
        for(i=0;i<4;i++)
        {
            R[c++]=Q[i];
        }
```

```
        System.out.println("Resultant Array:");
        for(i=0;i<10;i++)
        {
          System.out.print(R[i]+"\t");
        }

    }
}
```

57. Write a program to accept the year of graduation from school as an integer value from the user. Using the Binary Search technique on the sorted array of integers given below, output the message 'Record exists' if the value input is located in the array. If not, output the message Record does not exist".

(1982, 1987, 1993. 1996, 1999, 2003, 2006, 2007, 2009, 2010)

Ans.

```
import java.util.*;
class Sol57
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    int a[]={1982,1987,1993,1996, 1999, 2003, 2006, 2007, 2009, 2010},l,u,n,m,f=0;
    System.out.println("Enter the year of graduation:");
    n=sc.nextInt();
    l=0;u=a.length-1;
    while(l<=u)
    {
      m=(l+u)/2;
      if(a[m]==n)
      {
        f=1;
        break;
      }
      else if(a[m]>n)
        u=m-1;
      else
        l=m+1;
    }
    if(f==1)
      System.out.println("Record exists");
    else
      System.out.println("Record does not exist");
  }
}
```

58. Write a program to accept name and total marks of N number of students in two single subscript array name[] and totalmarks[].

Calculate and print:

i. The average of the total marks obtained by N number of students.

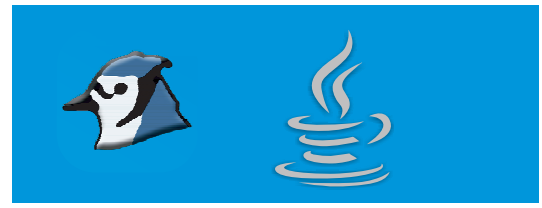
ii. Deviation of each student's total marks with the average.

[deviation=total marks of a student - average]

Ans.

```
import java.util.*;
class Sol58
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int N,i,s=0;
      System.out.println("Enter the no. of students:");
      N=sc.nextInt();
      String name[]=new String[N];
      int totalmarks[]=new int[N];
      float avg,dev;
      System.out.println("Enter the name and total marks:");
      for(i=0;i<N;i++)
      {
         name[i]=sc.nextLine();
         totalmarks[i]=sc.nextInt();
         s+=totalmarks[i];
      }
      avg=(float)s/N;
      System.out.println("Average marks obtained by the students:"+avg);
      System.out.println("Deviation of each student's total marks with the average");
      for(i=0;i<N;i++)
      {
         dev=totalmarks[i]-avg;
         System.out.println(name[i]+"\t\t"+dev);
      }
   }
}
```

Chapter 9
String Manipulation



Answer as directed

1. Give the output of the following program fragment:

```
String s=new String("He went to the market");
String r;
r=s.replace("went","is going");
System.out.println(r);
```

Ans.

Output:
He is going to the market

2. Give the output of the following program fragment:

```
String s="String";
int a=12,b=45;
System.out.println(s+a+b);
System.out.println(a+s+b);
System.out.println(a+b+s);
```

Ans.

Output:

```
String1245
12String45
57String
```

3. Give the output of the following program fragment:

```
String s="india",s1="IndIA",s2=s;
System.out.println(s.equals(s1));
System.out.println(s.equalsIgnoreCase(s1));
System.out.println(s2==s);
System.out.println(s.toUpperCase()==s1.toUpperCase());
System.out.println(s.startsWith("IN".toLowerCase()));
System.out.println(s1.endsWith("iA".toUpperCase()));
```

Ans.

Output:
false
true
true
false
true
true

4. What do the following functions return for:

```
String x =“hello”;
String y =“world”
System.out.println(x + y);
System.out.println(x.length();
System.out.println(x.charAt(3));
System.out.println(x.equals(y));
```

Ans.

Output:
helloworld
5
l
false

5. What is the output of the following:

(i)
```
System.out.println (“four :” + 4 + 2);
System.out.println (“four :”+(2+2));
```

(ii)
```
String S1 = “Hi”;
String S2 = “Hi”;
String S3 = “there”;
String S4 = “HI”;
System.out.println(S1 + “equals” + S2 + “→” + S1.equals(S2));
System.out.println(S1 + “equals” + S3 + “→” + S1.equals(S3));
System.out.println(S1 + “equals” + S4 + “→” + S1.equals(S4));
System.out.println(S1 + “equalsIgnoreCase” +S4 + “→” + S1.equalsIgnoreCase(S4));
```

Ans.

(i) four :42
four :4

(ii) Hi equals Hi→true
Hi equals there→false
Hi equals HI→false
Hi equalsIgnoreCase HI→true

6. If, String x = “Computer”;

String y = “Applications”;
What do the following functions return for:

(i) System.out.println(x.substring(1,5));
(ii) System out.println(x.indexOf(x.charAt(4)));
(iii) System.out.println(y+x.substring(5));
(iv) System.out.println(x.equals(y));

Ans.

(i) ompu (ii) 4
(iii) Applicationster (iv) false

7. What will be the output for the following program segment?

```
String s = new String("abc");
System.out.println(s.toUpperCase());
```

Ans.

ABC

8. What will be the output of the following code?

```
char x = 'A'; int m;
m=(x= ='a') ? 'A' : 'a';
System.out.println("m="+m);
```

Ans.

m=97

9. Write statements to show how finding the length of a character array and char[] differs from finding the length of a String object str.

Ans.

chars.length
str.length()

10. Write a statement each to perform the following task on a string:

(i) Find and display the position of the last space in a string s.
(ii) Convert a number stored in a string variable x to double data type

Ans.

(i) System.out.println(s.lastIndexOf(" "));
(ii) double d=Double.parseDouble(x);

11. Write a statement each to perform the following task on a string:

(i) Extract the second last character of a word stored in the variable wd.
(ii) Check if the second character of a string str is in uppercase.

Ans.

(i) char sl=wd.charAt(wd.length()-2);
(ii) if(Character.isUpperCase(str.charAt(1)))

12. Give the output of the following string functions:

(i) "ACHIEVEMENT".replace('E','A')
(ii) "DEDICATE".compareTo("DEVOTE")

Ans.

(i) ACHIAVAMANT
(ii) -18

13. Consider the following String array and give the output:

```
String arr[]={"DELHI", "CHENNAI", "MUMBAI", "LUCKNOW", "JAIPUR"};
System.out.println(arr[0].length()>arr[3].length());
System.out.print(arr[4].substring(0,3));
```

Ans.

false
JAI

14. String x[] = {“SAMSUMG”, “NOKIA”, “SONY”, “MICROMAX”, “BLACKBERRY”};

Give the output of the following statements:
(i) System.out.println(x[1]);
(ii) System.out.println(x[3].length());

Ans.

(i) NOKIA
(ii) 8

15. Write the output for the following:

String s=“Today is Test”;
System.out.println(s.indexOf(‘T’));
System.out.println(s.substring(0,7)+“ ”+“Holiday”);

Ans.

0
Today i Holiday

16. Give the ouput of the following string functions:

(i) “MISSISSIPPI”.indexOf(‘S’)+“MISSISSIPPI”.lastIndexOf(‘I’)
(ii) “CABLE”.compareTo(“CADET”)

Ans.

(i) 12
(ii) -2

17. State the output of the following program segment when executed:

String a = “Smartphone”, b = “Graphic Art”;
String h = a.substring(2, 5);
String k = b.substring(8).toUpperCase();
System.out.println(h);
System.out.println(k.equalsIgnoreCase(h));

Ans. art

True

18. State the output of the following program segment:

String str1 = “great”; String str2 = “minds”;
System.out.println(strl.substring(0,2).concat(str2.substring(1)));
System.out.println((“WH” + (strl.substring(2).toUpperCase())));

Ans.

grinds
WHEAT

19. State the value of characteristic and mantissa when the following code is executed.

String s = “4.3756”;

```
int n = s.indexOf('.');
int characteristic = Integer.parseInt(s.substring(0,n));
int mantissa = Integer.valueOf(s.substring(n+1));
```

Ans.

```
characteristic=4
mantissa=3756
```

20. State the output of the following program segment.

```
String s = "Examination";
int n = s.length();
System.out.println(s.startsWith(s.substring(5, n)));
System.out.println(s.charAt(2) == s.charAt(6));
```

Ans.

```
false
true
```

21. What will the following code output?

```
String s = "malayalam";
System.out.println(s.indexOf('m'));
System.out.println(s.lastIndexOf('m'));
```

Ans.

```
0
8
```

22. Give the output of the following:

```
String n = "Computer Knowledge";
String m = "Computer Applications";
System.out.println(n.substring (0,8).concat(m.substring(9)));
System.out.println(n.endsWith("e"));
```

Ans.

```
ComputerApplications
true
```

SECTION B

Write programs for the following:

1. Write a program to count the number of non-blank characters in a given sentence.

Ans.

```
import java.util.*;
class Sol1
{
   static void main()
```

```
    {
        Scanner sc=new Scanner(System.in);
        String s;
        int i,c=0;
        System.out.println("Enter a sentence:");
        s=sc.nextLine();
        for(i=0;i<s.length();i++)
        {
            char x=s.charAt(i);
            if (x!=' ')
                c++;
        }
        System.out.println("No. of non-blank character:"+c);
    }
}
```

2. Write a program to input a sentence and count the number of 'A' or 'a' in it.

Ans.

```
import java.util.*;
class Sol2
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        String s;
        int i,c=0;
        System.out.println("Enter a sentence:");
        s=sc.nextLine();
        for(i=0;i<s.length();i++)
        {
            char x=s.charAt(i);
            if (x=='A' || x=='a')
                c++;
        }
        System.out.println("No. of A or a:"+c);
    }
}
```

3. Write a program to input a sentence and count the number of vowels in it.

Ans.

```
import java.util.*;
class Sol3
{
    static void main()
    {
```

```
        Scanner sc=new Scanner(System.in);
        String s;
        int i,c=0;
        System.out.println("Enter a sentence:");
        s=sc.nextLine();
        s=s.toUpperCase();
        for(i=0;i<s.length();i++)
        {
          char x=s.charAt(i);
          if (x=='A' || x=='E' || x=='I' || x=='O' || x=='U')
            c++;
        }
        System.out.println("No. of vowels:"+c);
    }
}
```

4. Write a method to accept a word and print it in the following way:

Parameter->	TRIAL
Output->	L
	AL
	IAL
	RIAL
	TRIAL

Ans.

```
class Sol4
{
  static void main(String s)
  {
    int i,j;
    for(i=s.length()-1;i>=0;i--)
    {
      for(j=i;j<s.length();j++)
      {
        char x=s.charAt(j);
        System.out.print(x);
      }
      System.out.println();
    }
  }
}
```

5. Write a method to accept a name as parameter and print its initials.

Example,

Parameter->	AJAY PRATAP SINGH
Output->	A.P.S.

Ans.

```
class Sol5
{
   static void main(String n)
   {
      int i;
      n=" "+n.trim();
      n=n.toUpperCase();
      for(i=0;i<n.length()-1;i++)
      {
         if (n.charAt(i)==' ' && n.charAt(i+1)!=' ')
            System.out.print(n.charAt(i+1)+".");
      }
   }
}
```

6. Write a method to accept a name as parameter and print the initial first and then the title.

Example,

Parameter-> AJAY PRATAP SINGH RATHORE

Output-> A.P.S. RATHORE

Ans.

```
class Sol6
{
   static void main(String n)
   {
      int i,lp;
      n=" "+n.trim();
      n=n.toUpperCase();
      lp=n.lastIndexOf(' ');
      for(i=0;i<lp;i++)
      {
         if (n.charAt(i)==' '&& n.charAt(i+1)!=' ')
            System.out.print(n.charAt(i+1)+".");
      }
      System.out.print(n.substring(lp+1));
   }
}
```

7. Write a program to input a string and print out the text with the uppercase and lowercase letters reversed, but all other characters should remain the same as before.

Example :

INPUT : WelComE TO School

OUTPUT : wELcOMe to sCHOOL

Ans.

```
import java.util.*;
```

```
class Sol7
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s,n=" ";
      int i,c=0;
      System.out.println("Enter a sentence:");
      s=sc.nextLine();
      for(i=0;i<s.length();i++)
      {
         char x=s.charAt(i);
         if(x>='A' && x<='Z')
            x=(char)(x+32);
         else if(x>='a' && x<='z')
            x=(char)(x-32);
         n=n+x;
      }
      System.out.println("New sentence:"+n);
   }
}
```

8. Write a program to input a sentence and display only those words which begins with a vowel.

Ans.

```
import java.util.*;
class Sol8
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s,w=" ";
      int i,c=0;
      System.out.println("Enter a sentence:");
      s=sc.nextLine().toUpperCase();
      s=s.trim()+ " ";
      for(i=0;i<s.length();i++)
      {
         char x=s.charAt(i);
         if(x!=' ')
            w+=x;
         else
         {
            char y=w.charAt(0);
            if(y=='A' || y=='E' || y=='I' || y=='O' || y=='U')
               System.out.print(w+" ");
```

```
            w=" ";
          }
        }
      }
    }
```

9. Write a program to input a sentence and display only those words which begin and end with the same alphabet.

Ans.

```
import java.util.*;
class Sol9
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    String s,w=" ";
    int i;
    System.out.println("Enter a sentence:");
    s=sc.nextLine().toUpperCase();
    s=s.trim()+" ";
    for(i=0;i<s.length();i++)
    {
      char x=s.charAt(i);
      if(x!=' ')
        w+=x;
      else
      {
        char f=w.charAt(0);
        char l=w.charAt(w.length()-1);
        if(f==l)
          System.out.print(w+" ");
        w=" ";
      }
    }
  }
}
```

10. Write a program to input a sentence and print only those words which as all the five vowels present in it (i.e.A, E, I, O, U), in a given sentence.

For example,
INPUT
Enter a sentence: EDUCATION IS A MUST FOR THE DEVELOPMENT OF THE COUNTRY
OUTPUT
EDUCATION

Ans.

```
import java.util.*;
class Sol10
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s,w=" ";
      int i,fa=0,fe=0,fi=0,fo=0,fu=0;
      System.out.println("Enter a sentence:");
      s=sc.nextLine().toUpperCase();
      s=s.trim()+" ";
      for(i=0;i<s.length();i++)
      {
         char x=s.charAt(i);
         if(x!=' ')
         {
            w+=x;
            if(x=='A') fa=1;
            if(x=='E') fe=1;
            if(x=='I') fi=1;
            if(x=='O') fo=1;
            if(x=='U') fu=1;
         }
         else
         {
            if(fa+fe+fi+fo+fu==5)
               System.out.print(w+" ");
            w=" ";
         }
      }
   }
}
```

11. Write a program using a method Palin( ), to check whether a string is a Palindrome or not. A Palindrome is a string that reads the same from left to right and vice versa.

E.g. MADAM, ARORA, ABBA, etc.

Ans.

```
class Sol11
{
   static void Palin(String s)
   {
      int i;
      s=s.trim();
```

```
        String r=" ";
        for(i=0;i<s.length();i++)
        {
          r=s.charAt(i)+r;
        }
        if(s.equalsIgnoreCase(r))
          System.out.print("Palindrome");
        else
          System.out.print("Not Palindrome");
      }
}
```

12. Write a program to input any given string to calculate the total number of characters and vowels present in the string and also reverse the string :

Example : INPUT
Enter String : SNOWY

OUTPUT
Total number of characters : 05
Number of Vowels : 01
Reverse string : YWONS

Ans.

```
class Sol12
{
  static void main(String s)
  {
    int i,c=0;
    String r= " ";
    s=s.toUpperCase();
    for(i=0;i<s.length();i++)
    {
      char x=s.charAt(i);
      if (x== 'A' || x=='E' || x=='I' || x=='O' || x=='U')
        c++;
      r=x+r;
    }
    System.out.println("Total number of characters:"+c);
    System.out.println("Number of Vowels :"+c);
    System.out.println("Reverse string :"+r);
  }
}
```

13. Write a program to enter a sentence from the keyboard and count the number of times a particular word occurs in it. Display the frequency of the search word. Example:

INPUT:
Enter a sentence : the quick brown fox jumps over the lazy dog.

```
Enter a word to be searched   :   the
OUTPUT                        :
Searched word occurs          :   2 times.
```

Ans.

```
import java.util.*;
class Sol13
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    String s,w=" ",n;
    int i,c=0;
    System.out.println("Enter a sentence:");
    s=sc.nextLine().toUpperCase();
    s=s.trim()+" ";
    System.out.println("Enter a word to be searched:");
    n=sc.next();
    for(i=0;i<s.length();i++)
    {
      char x=s.charAt(i);
      if(x!=' ')
        w+=x;
      else
      {
        if(w.equals(n))
          c++;
        w= " ";
      }
    }
    System.out.println("Searched word occurs:"+c+"times");
  }
}
```

14. Write a program to input a sentence and print the number of characters found in the longest word of the given sentence. For example if S= "India is my country" then the output should be 7.

Ans.

```
import java.util.*;
class Sol14
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    String s,w=" ",l=" ";
    int i;
```

```
        System.out.println("Enter a sentence:");
        s=sc.nextLine().toUpperCase();
        s=s.trim()+ " ";
        for(i=0;i<s.length();i++)
        {
          char x=s.charAt(i);
          if(x!= ' ')
            w+=x;
          else
          {
            if(w.length()>l.length())
              l=w;
            w= " ";
          }
        }
        System.out.println("Length of the longest word:"+l.length());
    }
}
```

15. Design a class to overload a function num_calc( ) as follows:

a. void num_calc(int num, char ch) with one integer argument and one character argument, computes the square of integer argument if choice ch is 's' otherwise finds its cube.

b. void num_calc(int a, int b, char ch) with two integer arguments and one character argument. It computes the product of integer arguments if ch is 'p' else adds the integer.

c. void num_calc(String s1, String s2) with two string arguments, which prints whether the strings are equal or not.

Ans.

```
import java.util.*;
class Overlaod
{
  static  void num_calc(int num, char ch)
  {
    int c;
    if (ch=='s')
      c=num*num;
    else
      c=num*num*num;
    System.out.println("Answer:"+c);
  }
  static void num_calc(int a, int b, char ch)
  {
    int c;
    if (ch=='p')
      c=a*b;
    else
```

```
        c=a+b;
      System.out.println("Answer:"+c);
    }
    static  void num_calc(String s1, String s2)
    {
      if (s1.equalsIgnoreCase(s2))
        System.out.println("Same");
      else
        System.out.println("Not Same");
    }
}
```

16. In a set of 50 names, it is intended to find the total number of names which contains at least one pair of consecutive letters, e.g., SURABHI. In this 'A' and 'B' are consecutive letters and 'H' and 'I' are consecutive letters. Write a program for the above program.

Ans.

```
import java.util.*;
class Sol16
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s[]=new String[50];
      int i,c=0,j;
      System.out.println("Enter 50 names:");
      for(i=0;i<50;i++)
        s[i]=sc.nextLine();
      for(i=0;i<50;i++)
      {
        for(j=0;j<s[i].length()-1;j++)
        {
          if((char)(s[i].charAt(j)+1)==s[i].charAt(j))
          {
            c++;
            break;
          }
        }
      }
      System.out.println("No. of words :"+c);
   }
}
```

17. Write a program to input a line of text consisting of sentences terminated by either "." or "!" or "?". The words in the sentences may be separated by multiple spaces. The program should output:

a. The total number of words and
b. The number of alphabets in the given text.

Ans.

```
import java.util.*;
class Sol17
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s,w= " ";
      int i,c=0,p=0;
      System.out.println("Enter a string:");
      s=sc.nextLine();
      s=s.trim()+" ";
      for(i=0;i<s.length();i++)
      {
         char x=s.charAt(i);
         if(x!=' ' && x!='.' && x!='!' && x!='?')
         {
            w+=x;
            if(Character.isLetter(x))
               c++;
         }
         else
         {
            if(w!=" ")
               p++;
            w=" ";
         }
      }
      System.out.println("The total number of words :"+p);
      System.out.println("The number of alphabets in the given text. :"+c);
   }
}
```

18. Write a program to accept any string: Count and print the number of pairs of consecutive letters present in words in the forward direction only. Assume that all the letters in the string are in the same case, consecutive letters in two different words are not counted and 'za' or 'ZA' in any word is not a consecutive pair.

For example:
INPUT:

Enter String: ABSTRACT STRING IS BEING COUNTED EDUCATIONALLY.
OUTPUT:
Pairs of consecutive letter: 3

Ans.

```
import java.util.*;
class Sol18
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s;
      int i,c=0;
      System.out.println("Enter a string:");
      s=sc.nextLine().toUpperCase();

      for(i=0;i<s.length()-1;i++)
      {
         if((char)(s.charAt(i)+1)== s.charAt(i+1))
            c++;
      }
      System.out.println("Pairs of consecutive letter :"+c);
   }
}
```

19. Pig Latin is a language game of alterations played in English. To form the Pig Latin form of an English word the initial consonant sound is transposed to the end of the word and an ay is affixed (for example, trash yields ash-tray and plunder yields under-play). Write a program to input a word and change it to Pig Latin.

Ans.

```
import java.util.*;
class Sol19
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s,n;
      int i;
      System.out.println("Enter a word:");
      s=sc.nextLine().toUpperCase();

      for(i=0;i<s.length();i++)
      {
         char x=s.charAt(i);
         if(x=='A' || x=='E' || x=='I' || x=='O' || x=='U')
```

```
            break;
        }
        n=s.substring(i)+s.substring(0,i)+"AY";
        System.out.println("Pig Latin:"+n);
    }
}
```

20. An anagram is a word or a phrase made by transposing the letters of another word or phrase; for example, "parliament" is an anagram of "partial men," and "software" is an anagram of "swear oft". Write a program that figures out whether one string is an anagram of another string. The program should ignore white space and punctuation.

Ans.

```
import java.util.*;
class Sol20
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        String s1,s2,n;
        int i,c,p,f=0;
        char x;
        System.out.println("Enter 2 words:");
        s1=sc.nextLine().toUpperCase();
        s2=sc.nextLine().toUpperCase();
        for(x='A';x<='Z';x++)
        {
            c=p=0;
            for(i=0;i<s1.length();i++)
            {
                if(s1.charAt(i)==x)
                    c++;
            }
            for(i=0;i<s2.length();i++)
            {
                if(s2.charAt(i)==x)
                    p++;
            }
            if(c!=p)
                f=1;
        }
        if(f==0)
            System.out.println("Anagram");
        else
            System.out.println("Not Anagram");
    }
}
```

21. Write a program to input a string in uppercase and print the frequency of each character.

INPUT : COMPUTER HARDWARE
OUTPUT :

CHARACTERS	FREQUENCY
A	2
C	1
D	1
E	2
H	1
M	1
P	1
R	3
T	1
U	1
W	1

Ans.

```
import java.util.*;
class Sol21
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s;
      int i,c;
      char x;
      System.out.println("Enter a string:");
      s=sc.nextLine().toUpperCase();
      System.out.println("CHARACTERS\t\tFREQUENCY");
      for(x='A';x<='Z';x++)
      {
         c=0;
         for(i=0;i<s.length();i++)
         {
            if(s.charAt(i)==x)
               c++;
         }
         if(c>0)
         System.out.println(x+"\t\t"+c);
      }
   }
}
```

22. Input two words and find out the set that will be formed by the intersection of the characters found in both the strings.

INPUT:
Enter the first word: TOPPLE
Enter the second word: CRIPPLE
OUTPUT:
Intersecting Set: PLE

Ans.

```
import java.util.*;
class Sol22
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s1,s2,n=" ";
      int i;
      char x;
      System.out.println("Enter 2 words:");
      s1=sc.nextLine().toUpperCase();
      s2=sc.nextLine().toUpperCase();
      for(i=0;i<s1.length();i++)
      {
         x=s1.charAt(i);
         if(s2.indexOf(x)!=-1 && n.indexOf(x)==-1)
            n+=x;
      }
      System.out.println("Intersecting Set:"+n);
   }
}
```

23. Write a program to enter a sentence and a word from the keyboard and count the number of times a particular word occurs in it. Display the frequency of the search word.

Example:
INPUT:
Enter a sentence : the quick brown fox jumps over the lazy dog.
Enter a word to be searched: the
OUTPUT:
Searched word occurs: 2 times.

Ans.

```
import java.util.*;
class Sol23
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s,w=" ",n;
```

```
        int i,c=0;
        System.out.println("Enter a sentence:");
        s=sc.nextLine().toUpperCase();
        s=s.trim()+ " ";
        System.out.println("Enter a word to be searched:");
        n=sc.next();
        for(i=0;i<s.length();i++)
        {
          char x=s.charAt(i);
          if(x!= ' ')
            w+=x;
          else
          {
            if(w.equals(n))
              c++;
            w=" ";
          }
        }
        System.out.println("Searched word occurs :"+c+"times");
      }
    }
```

24. Write a program in Java to accept a string in lower case and change the first letter of every word to upper case. Display the new string.

Sample input: we are in cyber world
Sample output: We Are In Cyber World

Ans.

```
import java.util.*;
class Sol24
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    String s,n=" ";
    int i,c=0;
    System.out.println("Enter a sentence:");
    s=sc.nextLine();
    for(i=0;i<s.length();i++)
    {
        char x=s.charAt(i);
        if(x>='A' && x<='Z')
        x=(char)(x+32);
        else if(x>='a' && x<='z')
        x=(char)(x-32);
        n=n+x;
```

```
        }
        System.out.println("New sentence:"+n);
    }
}
```

25. Design a class to overload a function check() as follows:

(i) void check(String str, char ch) - to find and print the frequency of a character in a string.

Example:

Input:	Output:
str="success"	number of s present is =3
ch='s'	

(ii) void check(String s1) - to display only vowels from string s1. after converting it to lower case.

Example:

Input:	Output:
s1="computer"	o u e

Ans.

```
import java.util.*;
class Sol25
{
   static void  check(String str, char ch)
   {
     int i,c=0;
     for(i=0;i<str.length();i++)
     {
         char x=str.charAt(i);
         if(x==ch)
            c++;
     }
     System.out.println("number of"+ch+ "present is ="+c );
   }
   static void check(String s1)
   {
     s1=s1.toLowerCase();
     int i;
     for(i=0;i<s1.length();i++)
     {
        char x=s1.charAt(i);
        if(x=='a' || x=='e' || x=='i' || x=='o' || x=='u')
           System.out.print(x+ " ");
     }
   }
}
```

26. Write a program to input forty words in an array. Arrange these words in descending order of alphabets, using selection sort technique. Print the sorted array.

Ans.

```
import java.util.*;
class Sol26
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String a[]=new String[40],s;
      int i,p,j;
      System.out.println("Enter 40 words:");
      for(i=0;i<40;i++)
         a[i]=sc.nextLine();
      for(i=0;i<39;i++)
      {
         s=a[i];p=i;
         for(j=i+1;j<40;j++)
         {
            if(a[j].compareToIgnoreCase(s)<0)
            {
               s=a[j];
               p=j;
            }
         }
         a[p]=a[i];
         a[i]=s;
      }
      for(i=0;i<40;i++)
         System.out.println(a[i]);
   }
}
```

27. Special words are those words which starts and ends with the same letter.

Examples:
EXISTENCE
COMIC
WINDOW
Palindrome words are those words which read the same from left to right and vice-versa.
Example:
MALAYALAM
MADAM
LEVEL
ROTATOR
CIVIC

All palindromes are special words, but all special words are not palindromes.

Write a program to accept a word check and print whether the word is a palindrome or only special word.

Ans.

```
import java.util.*;
class Sol27
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String s,r=" ";
      int i;
      System.out.println("Enter a word:");
      s=sc.nextLine().toUpperCase();
      for(i=0;i<s.length();i++)
      {
         char x=s.charAt(i);
         r=x+r;
      }
        if(r.equals(s))
        System.out.println("Palindrome");
        else if(s.charAt(0)==r.charAt(0))
        System.out.println("Special");
        else
        System.out.println("Neither Palindrome nor special");
      }
}
```

28. Write a program to initialize the seven Wonders of the World along with their locations in two different arrays. Search for a name of the country input by the user. If found, display the name of the country along with its Wonder, otherwise display "Sorry Not Found!"

Seven wonders – CHICHEN ITZA, CHRIST THE RDEEEMER, TAJMAHAL, GREAT WALL OF CHINA, MACHU PICCHU, PETRA, COLOSSEUM
Locations – MEXICO, BRAZIL, INDIA, CHINA, PERU, JORDAN, ITALY
Example – Country Name: INDIA Output: INDIA – TAJMAHAL
Country Name: USA Output: Sorry Not Found!

Ans.

```
import java.util.*;
class Sol28
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String wonders[]={"CHICHEN ITZA","CHRIST THE RDEEEMER","TAJMAHAL",
```

```
                "GREAT WALL OF CHINA","MACHU PICCHU","PETRA","COLOSSEUM"};
        String locations[]={"MEXICO","BRAZIL","INDIA","CHINA","PERU","JORDAN","ITALY"};
        String c=" ";
        int i,f=0;
        System.out.println("Enter a country:");
        c=sc.nextLine().toUpperCase();
        for(i=0;i<locations.length;i++)
        {
          if(locations[i].equals(c))
          {
            System.out.println(c+"-"+wonders[i]);
            f=1;
          }
        }
        if(f==0)
          System.out.println("Sorry Not Found!");
    }
}
```

29. Design a class to overload a function Joystring() as follows:

(i) void Joystring(String s, char ch1, char ch2) with one string and two character arguments that replaces the character argument ch1 with the character argument ch2 in the given string s and prints the new string

Example:

Input value of s = "TECHNALAGY"

ch1 = 'A'

ch2 = 'O'

Output : "TECHNOLOGY"

(ii) void Joystring(String s) with one string argument that prints the position of the first space and the last space of the given String s.

Example:

Input value of = "Cloud computing means Internet based computing"

First Index : 5

Last Index : 36

(iii) void Joystring(String s1, String s2) with two string arguments that combines the two strings with a space between them and prints the resultant string

Example :

Input value of s1 = "COMMON WEALTH"

s2 = "GAMES"

Output : "COMMON WEALTH GAMES"

(use library functions)

Ans.

```
class Sol29
{
  void Joystring(String s, char ch1, char ch2)
```

```
    {
       s=s.replace(ch1,ch2);
       System.out.println(" Sorry Not Found!");
    }
     void Joystring(String s)
     {
        int f,l;
        f=s.indexOf(' ');
        l=s.lastIndexOf(' ');
        System.out.println("First Index :"+f);
        System.out.println("Last Index :"+l);
     }
      void Joystring(String s1, String s2)
      {
         String s3;
         s3=s1+" "+s2;
         System.out.println(s3);
      }
}
```

30. Write a program to input twenty names in an array. Arrange these names in descending order of alphabets, using the bubble sort technique.

Ans.

```
import java.util.*;
class Sol30
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      String names[]=new String[20];
      String t=" ";
      int i,j;
      System.out.println("Enter 20 names:");
      for(i=0;i<20;i++)
         names[i]=sc.nextLine();
      for(i=19;i>0;i--)
      {
         for(j=0;j<i;j++)
         {
            if(names[j].compareToIgnoreCase(names[j+1])>0)
            {
               t=names[j];
               names[j]=names[j+1];
               names[j+1]=t;
            }
```

```
        }
      }
      System.out.println("Alphabetical order:");
      for(i=0;i<20;i++)
        System.out.println(names[i]);
    }
}
```

31. Write a program to assign a full path and file name as given below. Using library functions, extract and output the file path, file name and file extension separately as shown.

Input C:\Users\admin\Pictures\flower.jpg
Output Path: C:\Users\admin\Pictures\
File name: flower
Extension: jpg

Ans.

```
import java.util.*;
class Sol31
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    String s,path,fname,ext;
    int l,p;
    System.out.println("Entera full path name and file name:");
    s=sc.nextLine();
    l=s.lastIndexOf('\\');
    path=s.substring(0,l+1);
    p=s.lastIndexOf(".");
    fname=s.substring(l+1,p);
    ext=s.substring(p+1);
    System.out.println("Path:"+path);
    System.out.println("File name:"+fname);
    System.out.println("Extension:"+ext);
  }
}
```

32. Write a program to accept a string. Convert the string to uppercase. Count and output the number of double letter sequences that exist in the string.

Sample Input: "SHE WAS FEEDING THE LITTLE RABBIT WITH AN APPLE
Sample Output: 4
Ans.

```
import java.util.*;
class Sol32
{
  static void main()
```

```
    {
        Scanner sc=new Scanner(System.in);
        String s;
        int i,c=0;
        System.out.println("Enter a string:");
        s=sc.nextLine();
        for(i=0;i<s.length()-1;i++)
        {
            if(s.charAt(i)==s.charAt(i+1))
                c++;
        }
        System.out.println("Answer:"+c);
    }
}
```

33. Write a program to accept the names of 10 cities in a single dimension string array and their STD (Subscribers Trunk Dialing) codes in another single dimension integer array. Search for a name of a city input by the user in the list. If found, display "Search Successful" and print the name of the city along with its STD code, or else display the message "Search Unsuccessful, No such city in the list'.

Ans.

```
import java.util.*;
class Sol33
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        String cities[]=new String[10];
        int STD[]=new int[10];
        String c=" ";
        int i,j,f=0;
        System.out.println("Enter 20 cities and STD codes:");
        for(i=0;i<10;i++)
        {
            cities[i]=sc.nextLine();
            STD[i]=sc.nextInt();
        }
        System.out.println("Enter the name of a city to search:");
        c=sc.nextLine();
        for(i=0;i<10;i++)
        {
            if(c.equalsIgnoreCase(cities[i]))
            {
                System.out.println(" "Search Successful");
                System.out.println(c+"-"+STD[i]);
```

```
            f=1;
        }
    }
    if(f==0)
        System.out.println("Search Unsuccessful, No such city in the list");
  }
}
```

34. Write a program to input a sentence and print the sentence in reverse order without reversing each word.

For example, If input is
He went to the market
Output should be
market the to went He

Ans.

```
import java.util.*;
class Sol34
{
  static void main()
  {
    Scanner sc=new Scanner(System.in);
    String s,w=" ",r=" ";
    char x;
    int i;
    System.out.println("Enter a sentence:");
    s=sc.nextLine();
    s=s.trim()+" ";
    for(i=0;i<s.length();i++)
    {
      x=s.charAt(i);
      if(x!= ' ')
        w=w+x;
      else
      {
        r=w+" "+r;
        w=" ";
      }
    }
    System.out.println("New Sentence:"+r);
  }
}
```

Model Test Paper-1

Question 1

(a) Differentiate between class variable and instance variable.

(b) Mention two difference between Binary search and Linear search.

(c) Define wrapper class.

(d) What do you mean by rvalue and lvalue?

(e) Why Java do not require a destructor?

Ans.

(a) Class variables only have one copy that is shared by all the different objects of a class, whereas every object has it's own personal copy of an instance.

(b) The major difference between linear search and binary search is that binary search takes less time to search an element from the sorted list of elements. So it is inferred that efficiency of binary search method is greater than linear search.
Another difference between the two is that there is a prerequisite for the binary search, i.e., the elements must be sorted while in linear search there is no such prerequisite.

(c) A Wrapper class is a class whose object wraps or contains a primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

(d) "lvalue" either means "expression which can be placed on the left-hand side of the assignment operator", or means "expression which has a memory address". "rvalue" is defined as "all other expressions".

(e) Java has its own garbage collection implementation so it does not require any destructor like C++ . Still we can have destructor along with garbage collector where developer can free resources and which can save garbage collector's work.

Question 2

(a) State the difference between Constructor and Method.

(b) Define fall through.

(c) What is the difference between ordinary compilation and Java compilation?

(d) What is the difference between String and Array?

(e) What is meant by private visibility of a method?

Ans. (a)

Java Constructor	Java Method
Constructor is used to initialize the state of an object.	Method is used to expose behaviour of an object.

Constructor must not have return type.	Method must have return type.
Constructor is invoked implicitly.	Method is invoked explicitly.
The java compiler provides a default constructor if you don't have any constructor.	Method is not provided by compiler in any case.
Constructor name must be same as the class name	Method name may or may not be same as class name.

(b) A "fall through" is what we call it when a case in a switch statement doesn't end in a break, return, throw, or any other control-breaking statement. In these cases, program execution continues to the next case block (regardless of the value in the switch), and so control "falls through" to the case below it.

(c) In ordinary compilation the source code is directly converted into machine or object code using an interpreter or a compiler .In Java compilation into an the source code is first converted into an intermediate code called the byte code.

(d) An array is used to hold multiple values of the same type. An array may hold

- multiple primitive values of same type, or
- multiple object references of same type.

String is a nothing but a sequence of characters within double quotes.

(e) Private access modifier simply means that the method on which it is used will be accessible only in the enclosing class.

Question 3

(a) Name the keyword that

(i) Transfers the control to the beginning of the loop and skips the rest of the loop body.

(ii) The variables whose single copy is made for all the objects of a class.

(iii) The keyword that is used to distinguish between instance variable and local variables.

(b) Give the output of the following:

```
int x=10;
int y=20;
if((x<y)||(x=5)>10)
System.out.print(y);
else
System.out.print(x);
```

(c) What will be the output of the following program segment?

```
String name[]={"Board","Examinations"};
System.out.println(name[0].length());
System.out.println(name.length());
```

(d) Find the output

```
class Aa
{
void display()
```

```
{
int a[]={2,5,8,5,3};
int i=2;
a[i]+=(a[i+1]++)-(--a[i-1]);
for(i=0;i<5;i++)
System.out.println(a[i]);
}
```

(e) Differentiate between base and derived class.

(f) Write the java expressions for

(g) Write function prototype that returns a Boolean value and takes two characters as parameter.

(h) In Java various Math functions are defined in which class and under which package.

(i) Write a single line statement using ternary operator to find the maximum of three numbers.

Ans.

(a) i) continue
ii) static or class variable
iii) this

(b) 20

(c) 5
2

(d) 2
4
9
6
3

(e) Derived class is the class to which data members are inherit or you can say that it is a child class. Derivation allows you to derive a class, called a derived class, from another class, called a base class.

(f) x=(-b+Math.sqrt(b*b-4*a*c)/(2*a);

(g) boolean function(char a, char b)

(h) class-Math Package- java.lang

(i) System.out.println((a>b)?(a>c)?a:c : (b>c)?b:c);

Question 4

Define a class Salary in Java with the following details:

Data members:

name[] array to store names of 20 employees.

netsal[] array to store net salary of 20 employees.

Member functions:

void input() to accept names of 20 employees and their corresponding net salaries.

void selection() to sort the array netsal in descending order and arrange the names accordingly.

Use selection sort technique.

void print() to print the sorted data in the following format.

Net Salary Name

…… ……

…… ……

Ans.

```
import java.util.*;
class Salary
{
   String name[]=new String[20];
   float netsal[]=new float[20];
   void input()
   {
      Scanner sc=new Scanner(System.in);
      int i;
      System.out.println("Enter the details:");
      for(i=0;i<20;i++)
      {
         name[i]=sc.nextLine();
         netsal[i]=sc.nextFloat();
      }
   }
   void selection()
   {
      int i,j,p;
      float ns;
      String t;
      for(i=0;i<19;i++)
      {
         ns=netsal[i];p=i;
         for(j=i+1;j<20;j++)
         {
            if(netsal[j]<ns)
            {
               ns=netsal[j];
               p=j;
            }
         }
         netsal[p]=netsal[i];
         netsal[i]=ns;
         t=name[i];
         name[i]=name[p];
         name[p]=t;
      }
   }
   void print()
```

```
    {
        int i,j;
        System.out.println("Net Salary\t\tName");
        for(i=0;i<20;i++)
        {
            System.out.println(netsal[i]+"\t\t"+name[i]);
        }
    }
}
```

Question 5

Write a program to accept any string: Count and print the number of pairs of consecutive letters present in words in the forward direction only. Assume that all the letters in the string are in the same case, consecutive letters in two different words are not counted and 'za' or 'ZA' in any word is not a consecutive pair.

For example:
INPUT:
Enter String: ABSTRACT STRING IS BEING COUNTED EDUCATIONALLY.
OUTPUT:
Pairs of consecutive letter: 3

Ans.

```
import java.util.*;
class Q5
{
  static void main()
  {
     Scanner sc=new Scanner(System.in);
     int i,c=0;
     char x,y;
     String s;
     System.out.println("Enter the sentence:");
     s=sc.nextLine();
     for(i=0;i<s.length()-1;i++)
     {
       x=s.charAt(i);
       y=s.charAt(i+1);
       if((char)(x+1)==y)
        c++;
     }
     System.out.println("No. of pairs:"+c);
  }
}
```

Question 6

Write a program to store following numbers in an array and shift all negative elements of an array to right hand side of the array without altering the original sequence (without another array).

i.e. if array contains: 5, -4, 3, -2, 6, -11, 12, -8, 9
Then output will be: 5, 3, 6, 12, 9, -4, -2, -11, -8

Ans.

```
class Q6
{
  static void main()
  {
    int i,a[]={5, -4, 3, -2, 6, -11, 12, -8, 9 },j,t;
    for(i=a.length-1;i>0;i--)
    {
      for(j=0;j<i;j++)
      {
        if (a[j]<0 && a[j+1]>0)
        {
          t=a[j];
          a[j]=a[j+1];
          a[j+1]=t;
        }
      }
    }
    for(i=0;i<a.length;i++)
      System.out.print(a[i]+" ");
  }
}
```

Question 7

Write a menu driven program using function overloading (the function name is display) to perform the
following:

(a) To find the sum of the series up to the limit given by user i.e 2+3+5+7+11+.....

(b) To print the following pattern:

000
001
010
011
100

101
110
111

Ans.

```
class Q7
{
  static void display(int n)
   {
     int i,j,s=0,c;
     for(i=1;i<=n;i++)
     {
        c=0;
       for(j=0;j<=i;j++)
       {
          if (i%j==0)
          c++;
        }
        if(c==2)
          s+=i;
     }
     System.out.print("Sum="+s);
   }
   static void display()
   {
     int i,j,d;
     String s=" ";
     for(i=1;i<=7;i++)
     {
       s="";
        for(j=i;j>0;j/=2)
        {
          d=j%2;
          s=d+s;
        }
        if(s.length()==1)
          s="00"+s;
        else if(s.length()==2)
          s="0"+s;
        System.out.println(s);
     }
   }
}
```

Question 8

Write a program to store n number of state name and its capital in ascending order and then input a state name and print its capital if it is in the list otherwise print the appropriate message using binary search.

Ans.

```
import java.util.*;
class Q8
{
   static void main()
   {
      String st[]={"Andhra Pradesh","Assam","Bihar","Chhattisgarh","Goa","Gujrat"};
      String cap[]={"Hyderabad","Dispur","Patna","Raipur","Panaji","Gandhinagar"};
      Scanner sc=new Scanner(System.in);
      String s;
      System.out.println("Enter the state name to search:");
      s=sc.nextLine();
      int l=0,u=st.length-1,f=0,m;
      while(l<=u)
      {
         m=(l+u)/2;
         if(st[m].equals(s))
         {
            f=1;
            System.out.println(cap[m]);
            break;
         }
         else if(s.compareTo(st[m])<0)
            u=m-1;
         else
            l=m+1;
      }
      if(f==0)
         System.out.println("Not found");
   }
}
```

Question 9

(a) WAP to calculate the sum of the following series.
S = 1/2! – 2/5! – 4/10! + 8/17! – 16/26! – 32/37! + 64/50! ……..n terms

(b) WAP to display the chain of 5 consecutive composite numbers within 1 to 100.
e.g 24, 25, 26, 27, 28
62, 63, 64, 65, 66

****

Ans.

(a)

```
import java.util.*;
class Q9a
{
   static void main()
   {
     Scanner sc=new Scanner(System.in);
     int n,i,p,m=2,c=3,v=1,j,b=0;
     double s=0;
     System.out.println("Enter the no. of terms:");
     n=sc.nextInt();
     for(i=1;i<=n;i++)
     {
        p=1;
        for(j=1;j<=m;j++)
            p=p*j;
        m=m+c;
        c=c+2;
        if(b%3==0)
         s=s+(double)v/p;
        else
         s=s-(double)v/p;
        v=2*v;b++;
      }
      System.out.println("Sum="+s);
   }
}
```

(b)

```
import java.util.*;
class Q9b
{
   static void main()
   {
     int n,i,j,c,f=0,m=0,p=0;
     for(i=1;i<=100;i++)
     {
        c=0;
        for(j=1;j<=i;j++)
```

```
        {
          if(i%j==0)
           c++;
         }
         if(c>2)
         {
           m++;
           if(m==1)
             p=i;
           if(m==5)
           {
             for(j=p;j<p+5;j++)
                System.out.print(j+" ");
             m=0;
             System.out.println();
           }
         }
         else
           m=0;
      }

    }
}
```

Model Test Paper-2

Question 1

(a) What do you know about instance variable?

(b) Explain the concept of try-catch-finally with a suitable example. What is the use of 'finally' keyword?

(c) State and explain the multiple branch selection statement provided by Java.

(d) Given numbers 23, 45, -33, 533, 100. Illustrate a binary search to locate the number 500.

(e) What is control variable?

Ans.

(a) Instance variable in java is used by Objects to store their states. Variables which are defined without the STATIC keyword and are Outside any method declaration are Object specific and are known as instance variables. They are called so because their values are instance specific and are not shared among instances.

(b) The try block encases any statements that might cause an exception to occur. The catch block(s) provide a place to handle the exception thrown by the statements within a try block.

The statements in the finally block are always executed. This is useful to clean up resources in the event of the try block executing without an exception and in the cases when there is an exception.

For example,

```
class x
{
   static void divide(int a,int b)
   {
      int c;
      try
      {
         c=a/b;
         System.out.println("Quotient:"+c);
      }
      catch(ArithmeticException e)
      {
         System.out.println("Division error");
      }
      finally
      {
```

```
            System.out.println("End of the program");
        }
    }
}
```

(c) The switch statement is a multi-way branch statement. The switch statement of Java is another selection statement that defines multiple paths of execution of a program. It provides a better alternative than a large series of if-else-if statements.

(d) Set of numbers: 23, 45, -33, 533, 100
Ascending Order: -33, 23, 45, 100, 533

Lower limit	Upper limit	Mid	Check
0	4	2	45<500
3	4	3	100<500
4	4	4	533>500
4	3	---	Not found

(e) A control variable in computer programming is a program variable that is used to regulate the flow of control of the program. In definite iteration, control variables are variables which are successively assigned (or bound to) values from a predetermined sequence of values.

Question 2

(a) Why is a class known as composite data type?

(b) What is Recursive method?

(c) What is Function Signature?

(d) Differentiate a constructor from a destructor.

(e) Which unit of a class gets called when the object of the class is created?

Ans.

(a) A composite data type is one which is composed with various primitive data type. A class defined with various primitive data types such as int, double etc; so it is known as a composite data type; and it is used to create objects which hold similar types of values and behaviours (functions).

(b) Recursion is a basic programming technique you can use in Java, in which a method calls itself to solve some problem. A method that uses this technique is recursive method.

(c) A function signature (or type signature, or method signature) defines input and output of functions or methods. A signature can include: parameters and their types. a return value and type. exceptions that might be thrown or passed back.

(d) Constructor is used to initialize the instance of a class. Destructor destroys the objects when they are no longer needed. Constructor is Called when new instance of a class is created. Destructor is called when instance of a class is deleted or released.

(e) Constructor

Question 3

(a) Encapsulation wraps up the attributes and methods of the object – comment.

(b) Write the prototype of the function that returns a double value and takes three double parameters x, y and z as its parameters.

(c) In the following statements assume a = 50, b = 10 and c = 20.

The statement is c + = (a>0 && a<=10)? + + a : a / b;

What is the value of c?

(d) A method definition is as shown below:

```
int power ( int x, int y )
{
int f = 1;
for ( int i = 1; i <= y; i + +)
f * = i;
return ( f );
}
```

(e) Answer the following:

i. What is the term given to the variables x and y?

ii. What is the significance of the type int used before the method name power?

Ans.

(a) Encapsulation is wrapping, just hiding properties and methods. Encapsulation is used for hide the code and data in a single unit to protect the data from the outside the world. Class is the best example of encapsulation.

(b) double function(double a, double b, double c)

(c) 25

(d) (i) Parameters or Arguments

(ii) The return type of the function is int.

Question 4

(a) Read the following overloaded method prototypes and answer the question that follow:

```
void calculate ( int x, double y );
void calculate ( double x, double y, double z);
```

(i) Write the java statement to invoke the method “calculate” using actual parameters a = 20, b = 45.52.

(ii) Write the java prototype to overload the method calculate to return a double result and accepts two double parameters.

(b) Write the output of the following:

```
int a[ ] = { 5, 8, 7, 3, 2 };
a[ 1 ] = a[ 3 ];
a[ 2 ] = a[ 4 ];
a[ 3 ] = a[ 1 ] + a[ 2 ];
for ( int i = 0; i < 5; i + + )
System.out.print( a[ i ] + "," );
```

(c) What are wrapper classes?

(d) What do you know about visibility control?

(e) What is an abstract method?

Ans.

(a) (i) calculate(a,b);
(ii) double calculate(double x,double y)

(b) 5.3.2.5.2.

(c) A Wrapper class is a class whose object wraps or contains a primitive data types. When we create an object to a wrapper class, it contains a field and in this field, we can store a primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

(d) Visibility control in Java is implemented using Access Modifiers. An Access Modifier is a key word in java that determines what level of access or visibility a particular java variable/ method or class has. There are 4 basic access modifiers in java. They are- Public, Protected, Default and Private.

Private is the most restrictive access modifier whereas public is the least restrictive. Default is the access protection you get when you do not specifically mention an access modifier to be used for a java object.

(e) Abstract methods, similar to methods within an interface, are declared without any implementation. They are declared with the purpose of having the child class provide implementation.

Question 5

Binomial co-efficient can be calculated by using the following formula:

$$nCm = \frac{n!}{m!\,(n-m)!}$$ [where! sign represents the factorial of a number]

WAP in java to calculate and print the binomial co-efficient of the given expression, taking the value n and m as input. Make use of the function int fact(int k), which returns the factorial of a number k.

Ans.

```
import java.util.*;
class Q5
{
   static int fact(int k)
   {
      int i,f=1;
      for(i=1;i<=k;i++)
         k=k*i;
      return k;
   }

   static void main()
   {
      int n,m;
      double C;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the value of n and m:");
      n=sc.nextInt();
      m=sc.nextInt();
      C=(double)fact(n)/(fact(m)*fact(n-m));
      System.out.println("Binomial Coefficient:"+C);
   }
}
```

Question 6

WAP in java for the following specification:

Class : Empl

Data Members : Emp_No, Name, Basic Salary, DA, HRA, TA, PF, Gross Salary

Member Methods:

get ( ) : To accept Employee No., Name and Basic Salary of the employees

calcu ( ) : To calculate the Gross Salary based on the following condition:

Basic Salary	DA(%)	TA(%)	HRA(%)	PF(%)
>=20,000	53	12	10	8
>=10,000 to <20,000	45	10	12	7.5
< 10,000	40	8	14	7

Gross Salary = (Basic Salary + DA + TA + HRA) – PF

display ( ) : To display the following data in given format:

EMPLOYEE No. NAME GROSS SALARY PF

Write a main method to create the object of the above class and call the above method to calculate and print the Employee No., Name, Gross Salary and PF of an employee.

Ans.

```
import java.util.*;
class Empl
{
    int Emp_No;
    String Name;
    double basic,DA, HRA, TA, PF, gross;
    void get()
    {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the details:");
      Emp_No=sc.nextInt();
      Name=sc.nextLine();
      basic=sc.nextDouble();
    }
    void calcu()
    {
      if(basic>=20000)
      {
        DA=53/100.0*basic;
        TA=12/100.0*basic;
        HRA=10/100.0*basic;
        PF=8/100.0*basic;
      }
      else if(basic>=10000 && basic<20000)
      {
        DA=45/100.0*basic;
        TA=10/100.0*basic;
        HRA=12/100.0*basic;
        PF=7.5/100.0*basic;
      }
      else
      {
        DA=40/100.0*basic;
        TA=8/100.0*basic;
        HRA=14/100.0*basic;
        PF=7/100.0*basic;
      }
      gross=basic+DA+TA+HRA-PF;
    }

    void display()
    {
      System.out.println("EMPLOYEE No.\t\tNAME\t\tGROSS SALARY\t\tPF");
```

```
        System.out.println(Emp_No+"\t\t"+Name+"\t\t"+gross+"\t\t"+PF);
    }

    public static void main(String args[])
    {
        Empl ob=new Empl();
        ob.get();
        ob.calcu();
        ob.display();
    }
}
```

Question 7

Input a number and find the nearest prime number to it.
Sample Input: 25
Sample Output: Nearest Prime number is 23

Ans.

```
import java.util.*;
class Q7
{
    static boolean isprime(int n)
    {
        int i,c=0;
        for(i=1;i<=n;i++)
        {
            if(n%i==0)
                c++;
        }
        if(c==2)
            return true;
        else
            return false;
    }

    static void main()
    {
        int a,p,n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number:");
        a=sc.nextInt();
        p=a-1;n=a+1;
        while(isprime(p)==false)
        {
            p--;
```

```
            if(p<0)
               break;
         }
         while(isprime(n)==false)
            n++;
         if(p>0)
         {
            if(a-p<n-a)
               System.out.println("Nearest Prime:"+p);
            else if(a-p>n-a)
               System.out.println("Nearest Prime:"+n);
         else
            System.out.println("Nearest Prime:"+n","+p);

         }
         else
            System.out.println("Nearest Prime:"+n);

      }
   }
```

Question 8

Write a menu driven program in java to perform the following (using switch case):
(i) to print the value of S where
S = 1/1! + (1 + 2)/(1! + 2!) + (1 + 2 + 3) / (1! + 2! + 3!) + ……. upto n terms.
(ii) to print the value of S where
S = 1n + 2(n-1) + ... + (n-1)2 + n1

Ans.

```
import java.util.*;
class Q8
{
   static long fact(long k)
   {
      long i,f=1;
      for(i=1;i<=k;i++)
         k=k*i;
      return k;
   }

   static void main()
   {
      int ch,n,i,a=0,j;
      long b=0;
```

```
        double s=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("MENU");
        System.out.println("1.first series");
        System.out.println("2.second series");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
           case 1:
              System.out.println("Enter the no. of terms:");
              n=sc.nextInt();
              for(i=1;i<=n;i++)
              {
                 a=a+i;
                 b=b+fact((long)i);
                 s=s+(double)a/b;
              }
              System.out.println("Sum="+s);
              break;
          case 2:
              System.out.println("Enter the value of n:");
              n=sc.nextInt();
              for(i=1,j=n;i<=n;i++,j--)
              {
                 b=b+i*j;
              }
              System.out.println("Sum="+b);
              break;
          default:
              System.out.println("Invalid choice");
        }
    }
}
```

Question 9

WAP in java to accept the NAME and TOTAL MARKS obtained in an exam of a class having N number of students and display the NAME and TOTAL MARKS of the students according to the rank 1st, 2nd and 3rd.

Ans.

```
import java.util.*;
class Q9
```

```
{

   static void main()
   {
      int N,i,j;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the no. of students:");
      N=sc.nextInt();
      int tm[]=new int[N],t1;
      String name[]=new String[N],t2;
      System.out.println("Enter the names and total marks:");
      for(i=0;i<N;i++)
      {
         name[i]=sc.nextLine();
         tm[i]=sc.nextInt();
         sc.nextLine();//dummy input
      }
      for(i=N-1;i>0;i--)
      {
         for(j=0;j<i;j++)
         {
            if(tm[j]<tm[j+1])
            {
               t1=tm[j];tm[j]=tm[j+1];tm[j+1]=t1;
               t2=name[j];name[j]=name[j+1];name[j+1]=t2;
            }
         }
      }
      for(i=0;i<N;i++)
      {
         System.out.println(name[i]+"\t\t"+tm[i]);
      }
   }
}
```

Model Test Paper-3

Question 1

(a) What is the difference between an object and a class?

(b) What does the token 'keyword' refer to, in the context of Java? Give an example for keyword.

(c) What is the difference between entry controlled and exit controlled loop?

(d) What are the two ways of invoking functions?

(e) What is the difference between / and % operators?

Ans.

(a) A class is a blueprint or template that defines the characteristics and behaviour of an entity. Object on the other hand is an instance of a class.

(b) A keyword refers to the reserve words that has a special meaning to the Java compiler. Eg, for, if, else, while etc.

(c) Loop, where test condition is checked before entering the loop body, known as Entry Controlled Loop. Loop, where test condition is checked after executing the loop body, known as Exit Controlled Loop.

(d) Call by Value and Call by Reference.

(e) / is used to find the quotient and % is used to find the remainder when 2 numbers are divided.

Question 2

(a) State the total size in bytes, of the arrays a[4] of char data type and p[4] of float data type.

(b) (i) Name the package that contains Scanner class.

(ii) Which unit of the class gets called, when the object of the class is created.

(c) Give the output of the following:

```
String n="Computer Kowledge";
String m="Computer Applications";
System.out.println(n.substring(0,8).concat(m.substring(9)));
System.out.println(n.endsWith("e"));
```

(d) Write the output of the following:

(i) System.out.println(Character.isUpperCase('R'));

(ii) System.out.println(Character.toUpperCase('j'));

(e) What is the role of keyword void in declaring functions?

Ans.

(a) a[4]= 8 bytes and p[4]=16 bytes

(b) (i) java.util

(ii) Constructor

(c) ComputerApplications

True

(d) (i) true

(ii) J

(e) It is used to ensure that a function do not return any value.

Question 3

(a) Analyse the following program segment and determine how many times the loop will be executed and what will be the output of the program segment?

```
int p=200;
while(true)
{
        if(p<100)
        break;
        p=p-20;
}
System.out.println(p);
```

(b) What will be the output of the following code?

(i)
```
int k=5, j=9;
k+ = k++ - ++j + k;
System.out.println("k="+k);
System.out.println("j="+j);
```

(ii)
```
double b=-15.6;
double a=Math.rint(Math.abs(b));
System.out.println("a="+a);
```

(c) Explain the concept of constructor overloading with an example.

(d) Give the prototype of a function search which receives a sentence sentnc and a word wrd and returns 1 or 0.

(e) Write an expression in Java for

(f) Write a statement each to perform the following task on a string:

(i) Find and display the position of the last space in a String s.

(ii) Convert a number stored in a String variable x to double data type.

(g) Name the keyword that

(i) informs that an error has occurred in an input/output operation.

(ii) distinguishes between instance variables and class variables.

(h) What are library classes? Give an example.

Ans.

(a) The loop executes 7 times. It will display 80 as output.

(b) (i) k=6
j=10

(ii) a=16.0

(c) In addition to overloading methods, we can also overload constructors in java. Overloaded constructor is called based upon the parameters specified when new is executed. Sometimes there is a need of initializing an object in different ways. This can be done using constructor overloading.

For example,

```
class Box
{
   double width, height, depth;

   // constructor used when all dimensions
   // specified
   Box(double w, double h, double d)
   {
      width = w;
      height = h;
      depth = d;
   }

   // constructor used when no dimensions
   // specified
   Box()
   {
      width = height = depth = 0;
   }

   // constructor used when cube is created
   Box(double len)
   {
      width = height = depth = len;
   }

   // compute and return volume
   double volume()
```

```
    {
        return width * height * depth;
    }
}
```

(d) int search(String sentc, String wrd)

(e) z=(5*x*x*x+2*y)/(x+y);

(f) (i) System.out.println(s.lastIndexOf(" ");

(ii) double d=Double.parseDouble(x);

(g) (i) try-catch

(ii) static

(h) The Java Class Library (JCL) is a set of dynamically loadable libraries that Java applications can call at run time. Because the Java Platform is not dependent on a specific operating system, applications cannot rely on any of the platform-native libraries. Instead, the Java Platform provides a comprehensive set of standard class libraries, containing the functions common to modern operating systems. Eg. Java.lang, java.util, java.awt, java,swing etc.

Question 4

Define a class called mobike with the following description:

Instance variables/ data members:

int bno	-	to store the bike's number
int phno	-	to store the phone number of the customer
String name	-	to store the name of the customer
int days	-	to store the number of days the bike is taken on rent.
int charge	-	to calculate and store the rental charge

Member Methods:

void input()	-	to input and store the detail of the customer
void compute()	-	to compute the rental charge.

The rent for a mobike is charged on the following rental basis:

First five days	₹ 500 per day.
Next five days	₹ 400 per day
Rest of the days	₹ 200 per day.

void display() to display the details in the following format:

Bike No.	Phone No.	Name	No. of days	Charge
_____	_____	_____	_____	_____

Ans.

```
import java.util.*;
class mobike
{
   int bno,phno,days,charge;
   String name;
```

```
    void input()
    {
       Scanner sc=new Scanner(System.in);
       System.out.println("Enter the details:");
       bno=sc.nextInt();
       name=sc.nextLine();
       phno=sc.nextInt();
       days=sc.nextInt();
    }
    void compute()
    {
       if (days<=5)
          charge=500*days;
       else if(days<=10)
          charge=500*5+(days-5)*400;
       else
          charge=500*5+5*400+(days-10)*200;
    }
    void display()
    {
       System.out.println("Bike No.\t\tPhone No.\t\tName\t\t No. of days\t\tCharge");
       System.out.println(bno+"\t\t"+phno+"\t\t"+name+"\t\t"+days+"\t\t"+charge);
    }
}
```

Question 5

Write a program to input and store the weight of ten people. Sort and display them in descending order using the selection sort technique.

Ans.

```
import java.util.*;
class Q5
{

   static void main()
   {
      float wt[]=new float[10],l;
      int i,j,p;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the weight of 10 people:");
      for(i=0;i<10;i++)
```

```
        {
            wt[i]=sc.nextFloat();
        }
        for(i=0;i<9;i++)
        {
            l=wt[i];p=i;
            for(j=i+1;j<10;j++)
            {
                if(wt[j]>l)
                {
                    l=wt[j];
                    p=j;
                }
            }
            wt[p]=wt[i];
            wt[i]=l;
        }
        for(i=0;i<10;i++)
        {
            System.out.println(wt[i]);
        }
    }
}
```

Question 6

Write a program to input a number and print whether the number is a special number or not. (A number is said to be special number, if the sum of the factorial of the digits of the number is same as the original number).

Example: 145 is a special number, because 1!+4!+5!=1+24+120=145

(Where ! stands for factorial of the number and the factorial value of a number is the product of all integers from 1 to that number, example 5!=1*2*3*4*5=120).

Ans.

```
import java.util.*;
class Q6
{
    static int fact(int k)
    {
        int i,f=1;
        for(i=1;i<=k;i++)
```

```
        k=k*i;
    return k;
  }

  static void main()
  {
    int n,d,i,s=0;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number:");
    n=sc.nextInt();
    for(i=n;i>0;i/=10)
    {
      d=i%10;
      s=s+fact(d);
    }
    if(s==n)
      System.out.println("Special No.");
    else
      System.out.println("Not a Special No.");
  }
}
```

Question 7

Write a program to accept a word and convert it into lowercase if it is in uppercase, and display the new word by replacing only the vowels with the character following it.

Example:

Sample Input: ComPuter

Sample output: cpmpvtfr

Ans.

```
import java.util.*;
class Q7
{
  static void main()
  {
    int i;
    String w,n=" ";
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a word:");
    w=sc.nextLine();
    w=w.toLowerCase();
    for(i=0;i<w.length();i++)
```

```
        {
            char x=w.charAt(i);
            if(x=='a' || x=='e' || x=='i' || x=='o' || x=='u')
            n=n+(char)(x+1);
            else
            n=n+x;
        }
        System.out.println("New word:"+n);
    }
}
```

Question 8

Design a class to overload a function compare( ) as follows:

(a) void compare(int, int) - to compare two integer values and print the greater of the two integers.

(b) void compare(char, char) - to compare the numeric value of two characters and print the character with higher numeric value.

(c) void compare(String, String) - to compare the length of the two strings and print the longer of the two.

Ans.

```
class Overload
{
    void compare(int a, int b)
    {
       if(a>b)
         System.out.println(a);
       else
         System.out.println(b);
    }
    void compare(char a, char b)
    {
        if(a>b)
        System.out.println(a);
        else
        System.out.println(b);
    }
    void compare(String a, String b)
    {
        if(a.length()>b.length())
```

```
            System.out.println(a);
            else
            System.out.println(b);
        }
    }
```

Question 9

Write a menu driven program to perform the following: (Use switch-case statement)

(i) To print the series 0,3,8,15,24… n terms(value of 'n' is to be an input by the user.)

(ii) To find the sum of the series given below:

S=1/2+3/4+5/6+7/8+--------19/20

Ans.

```
import java.util.*;
class Q9
{

   static void main()
   {
      int ch,n,i;
      float s=0;
      Scanner sc=new Scanner(System.in);
      System.out.println("MENU");
      System.out.println("1.first series");
      System.out.println("2.second series");
      System.out.println("Enter your choice:");
      ch=sc.nextInt();
      switch(ch)
      {
         case 1:
            System.out.println("Enter the no. of terms:");
            n=sc.nextInt();
            for(i=1;i<=n;i++)
            {
               System.out.print(((i*i)-1)+" ");
            }
            break;
        case 2:
            for(i=1;i<=19;i++)
```

```
            {
               s=s+(float)i/(i+1);
            }
            System.out.println("Sum="+s);
            break;
         default:
            System.out.println("Invalid choice");
      }
   }
}
```

Model Test Paper-4

Question 1

(a) Define Abstraction in Java.

(b) What are actual parameters and formal parameters?

(c) Differentiate between for loop and do-while loop.

(d) Define Wrapper class with an example.

(e) How much bytes does the following data types occupy:

(i) Short

(ii) long

(iii) double

(iv) char

Ans.

(a) Abstraction is a process of hiding the implementation details from the user. Only the functionality will be provided to the user. In Java, abstraction is achieved using abstract classes and interfaces.

(b) Java tutorial: actual parameter vs formal parameter. Formal parameters are the parameters as they are known in the function/method definition. Actual parameters are also known as arguments and are passed by the caller on method invocation (calling the method).

(c) Difference:

for loop	do while loop
It is an entry controlled loop.	It is an exit controlled loop.
The loop do not execute if the condition is false.	The loop executes at least once even if the condition is false.

(d) A Wrapper class is a class whose object wraps or contains a primitive data types. When we create an object to a wrapper class, it contains a field and in this field, we can store a primitive data types. In other words, we can wrap a primitive value into a wrapper class object. Example- Integer, Character Float, Short etc.

(e) (i) 2 bytes
(ii) 8 bytes
(iii) 8 bytes
(iv) 2 bytes

Question 2

(a) Declare the array of 5 integers and initialize first five even numbers to it.

(b) What is ternary operator? Give one example.

(c) What is meant by implicit and explicit type conversion?

(d) Name the branching statements in Java.

(e) Name any two packages in Java.

Ans.

(a) int a[]={2,4,6,8,10};

(b) The ternary operator is an operator that takes three arguments. The first argument is a comparison argument, the second is the result upon a true comparison, and the third is the result upon a false comparison. If it helps you can think of the operator as shortened way of writing an if-else statement. It is often used as a way to assign variables based on the result of an comparison. When used correctly it can help increase the readability and reduce the amount of lines in your code.
Example: b=(a>10)?15:16;

(c) When the type conversion is performed automatically by the compiler without the programmer's intervention, the type conversion is referred to as implicit type conversion. The compiler basically promotes all operands to the data type of the largest operand.

The type conversion which is enforced by the programmer is called explicit type conversion. Basically, the programmer forces an expression to be of a specific type. Explict type conversion is also called type casting.

(d) Branching statements: if, if-else, if-else if-else and switch-case.

(e) java.util and java.lang

Question 3

(a) What will be the output of the following program segment?

int a = 0, b = 30, c = 40;

a = - - b + c++ + b; System out printin ("a ="+ a);

(b) Suppose, String str = "STRING";

What is the result in each of the following cases?

(i) System.out.print(str.charAt(2) + " " + str.charAt(3));

(ii) System.out.print(str.substring(3,1).equals(str.substring(3,4)));

(iii) System.out.println (str.startsWith("ST"));

(iv) System.out.print (str.indexOf('R');

(c) State two differences in binary search and linear search techniques.

(d) State the output:

(i) x= y + 5; if x = 2 & y = 2

(ii) X = (a. b)? a; b; if a = 2, b = 5

(e) What is the default value of char, String, int and double type variables of java.

(f) Write the significance of using static variables with examples.

(g) Write the following conditions using ternary operator:

If amount is greater than 1000 and less than 10000, the commission is ₹ 4000 else it is 0.

(h) Consider the array: 13, 19, 6, 2, 35, 28, 5, 16, 65, 4 which sorting algorithm will produce the following result after 3 iterations? 13 6 2 19 35 28 5 16 65 4

(i) Define a counter variable in java.

(j) Explain the term package.

Ans.

(a) a = 98

(b) (i) R I
(ii) Error
(iii) true
(iv) 2

(c) Difference:

Linear Search	Binary Search
Array may not be sorted	Array should be sorted.
Searching is slow.	Searching is faster.

(d) (i) x=7
(ii) X=2

(e) char ='\0'
String=null
int=0
double=0.0

(f) A static variable is common to all the instances (or objects) of the class because it is a class level variable. In other words you can say that only a single copy of static variable is created and shared among all the instances of the class. Memory allocation for such variables only happens once when the class is loaded in the memory.

(g) Commission=(amount>1000 && amount<10000)?4000:0;

(h) Bubble Sorting

(i) A counter variable is a variable that is used to keep track of the number of times an operation is performed.

(j) Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces. Packages are used for:

- Preventing naming conflicts. For example there can be two classes with name Employee in two packages, college.staff.cse.Employee and college.staff.ee.Employee
- Making searching/locating and usage of classes, interfaces, enumerations and annotations easier
- Providing controlled access: protected and default have package level access control. A protected member is accessible by classes in the same package and its subclasses. A default member (without any access specifier) is accessible by classes in the same package only.
- Packages can be considered as data encapsulation (or data-hiding).

Question 4

Write a program to check if a number is a Magic Number or not. A number is said to be a Magic number if the sum of its digits are calculated till a single digit is obtained by recursively adding the sum of its digits. If the single digit comes to be 1 then the number is a magic number. Example- 199 is a magic number as 1+9+9=19 but 19 is not a single digit number so 1+9=10 and then 1+0=1 which is a single digit number and also 1. Hence it is a magic number.

Ans.

```
import java.util.*;
class Q4
{
   static void main()
   {
      Scanner sc=new Scanner(System.in);
      int n,s=0,d;
      System.out.println("Enter a number:");
      n=sc.nextInt();
      while(n>9)
      {
         s=0;
         while(n>0)
         {
            d=n%10;
            s+=d;
            n=n/10;
         }
         n=s;
      }
      if(n==1)
         System.out.println("Magic Number");
      else
         System.out.println("Not Magic Number");
   }
}
```

Question 5

Write a Java program to print the first 15 numbers of the Pell series.

In mathematics, the Pell numbers are an infinite sequence of integers. The Sequence of Pell numbers starts with 0 and 1, and then each Pell number is the sum of twice the previous Pell number and the Pell number before that.: thus, 70 is the companion to 29, and 70 = 2 x 29 + 12 = 58 + 12. The first few terms of the sequences are: 0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, 13860…

Ans.

```
import java.util.*;
class Q5
{
   static void main()
   {
      int i,f=1,s=0,t;
      for(i=1;i<=15;i++)
      {
         t=f+2*s;
         System.out.println(t+ " ");
         f=s;
         s=t;
      }
   }
}
```

Question 6

Maharashtra State Electricity board charges their consumers according to the units consumed (per month) as per the given tariff: UNITS CONSUMED CHARGES Up to 100 units 80 paise / unit More than 100 upto 200 units `. 1 / unit More than 200 units ` 1.25 / unit

In addition to the above mentioned charges, every consumer has to pay ` 50 as Service Charge per month and calculate the Electricity Bill. Write a program to create the object of the class and call the member methods.

Ans.

```
import java.util.*;
class Q6
{
   void calc()
   {
      Scanner sc=new Scanner(System.in);
      int units;
      double charge;
      System.out.println("Enter the no. of units:");
      units=sc.nextInt();
      if(units<=100)
      charge=units*0.80;
      else if(units<=200)
      charge=100*0.80+(units-100)*1.00;
      else
      charge=100*0.80+100*1.00+(units-200)*1.25;
      charge=50+charge;
```

```
        System.out.println("Charge:"+charge);
    }
    public static void main(String args[])
    {
        Q7 ob=new Q7();
        ob.calc();
    }
}
```

Question 8

Design a class to overload a function prStr() are as follows: void prStr(String s1, String s2) – Print the string that has more number of vowels from amongst s1 and s2.

void prStr(String s, char ch) – Replace all blank spaces from String s with ch and print the String s. void prStr(String s) – Print the first and last position of letter 'G' in String s.

Ans.

```
class Overload
{
    void prStr(String s1, String s2)
    {
        int i,c1=0,c2=0;
        s1=s1.toUpperCase();
        s2=s2.toUpperCase();
        for(i=0;i<s1.length();i++)
        {
            char x=s1.charAt(i);
            if (x=='A' || x=='E' || x=='I' || x=='O' || x=='U')
                c1++;
        }
        for(i=0;i<s2.length();i++)
        {
            char x=s2.charAt(i);
            if (x=='A' || x=='E' || x=='I' || x=='O' || x=='U')
            c2++;
        }
        if(c1>c2)
        System.out.println(s1);
        else
        System.out.println(s2);
    }
    void prStr(String s, char ch)
    {
        s=s.replace(' ', ch);
```

```
        System.out.println(s);
    }
    void prStr(String s)
    {
        int f,l;
        f=s.indexOf('G');
        l=s.lastIndexOf('G');
        System.out.println("First Occurence:"+f);
        System.out.println("Last Occurence:"+l);
    }
}
```

Question 9

Write a program using switch-case statements: 1. input an string array of size 10 and sort them in descending order using bubble sort technique. 2. initialize an array of size 9 Indian currency notes and initialize another array with their respective currency serial numbers. Search for a currency note input by the user, in the list.

If found, display "Search Successful" and print the currency along with the serial number, otherwise display "Search unsuccessful Name not enlisted." Write an appropriate message for incorrect option.

Ans.

```
import java.util.*;
class Q9
{

    static void main()
    {
        String w[]=new String[10],t;
        int i,j,ch,f=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("MENU");
        System.out.println("1. for sorting a string array");
        System.out.println("2. for searching a currency");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        switch(ch)
        {
            case 1:
                System.out.println("Enter 10 strings:");
                for(i=0;i<10;i++)
                {
                    w[i]=sc.nextLine();
                }
```

```
            for(i=9;i>0;i--)
            {
               for(j=0;j<i;j++)
               {
                  if(w[j].compareToIgnoreCase(w[j+1])>0)
                  {
                     t=w[j];
                     w[j]=w[j+1];
                     w[j+1]=t;
                  }
               }
            }
            for(i=0;i<10;i++)
            {
               System.out.println(w[i]);
            }
            break;
         case 2:
            int curr[]={2000,500,200,100,50,20,10,5,2},c;
            String ser[]={"12345AB","756DDA1","T5RRST45","ASSS3456","YUYU12GH",
                          "756DDA1","T5RRST45","ASSS3456","YUYU12GH"};
            System.out.println("Enter currency note:");
            c=sc.nextInt();
            for(i=0;i<curr.length;i++)
            {
               if(c==curr[i])
               {
                  System.out.println("Curreny note:"+c);
                  System.out.println("Serial:"+ser);
                  f=1;
                  break;
               }
            }
            if(f==0)
               System.out.println("Curreny note not found");
            break;
         default:
            System.out.println("Invalid choice");
      }
   }
}
```

Model Test Paper-5

Question 1

(a) Name any two library packages.

(b) State two main features of an Object Oriented Programming language.

(c) What is meant by a user defined data type?

(d) Show the use of logical operator && with an example.

(e) Explain the term - "pass by reference".

Ans.

(a) java.util and java.lang

(b) Encapsulation and Polymorphism

(c) A user-defined data type (UDT) is a data type that derived from an existing data type. You can use UDTs to extend the built-in types already available and create your own customized data types.

(d) A && logical operator returns a Boolean result that's based on the Boolean result of one or two other expressions. Returns true if both of the operands evaluate to true. Both operands are evaluated before the And operator is applied.

(e) When composite data types are passed to a method it is termed as pass by reference. Thus any changes made to the formal parameters are reflected back to the actual parameters.

Question 2

(a) Give an example of Syntax error.

(b) How is I/O Exception helpful in handling I/O errors?

(c) Differentiate between indexOf() and valueOf() methods.

(d) What is the scope of the keyword protected in accessing methods?

(e) Find the output of z:

```
int y = 14;
int z = ++y * (y– + –y);
```

Ans.

(a) C=a+b

(b) It is helpful when the expected data from the I/O device is not received for whatever reasons.

(c) The java string indexOf() method returns index of given character value or substring. The valueOf method returns the relevant Number Object holding the value of the argument passed.

(d) The protected methods are accessible by classes within the same package but only by sub-classes in other packages.

(e) z=450

Question 3

(a) Using an example, show how the switch statement works.

(b) Write an expression for the following:

Ö(a+b)

$f = \frac{-}{2} - d$

(c) Pick out the errors and correct them in the given code fragment:

```
class PickCode
{
private char ch;
private boolean n;
pickcode()
{
ch= 'C';
n == true;
}
}
```

(d) What are input and output streams?

(e) How is String implemented?

(f) Write the output for the following statements:

```
String strl ="Java is Fun";
String str2 = "amytime";
System.out.println(str2.replace(m','n'));
System.out.println(strl.concat(str2));
```

(g) Rewrite the program segment using multiple if-construct.

```
if(n%2==0 && n>35 && n<67)
{
a=a+4;
n=n/a;
rem=n%10;
 }
```

(h) Create an object for a class Park.

(i) What is a parameterized constructor?

(j) What is the use of static in the main method?

Ans.

(a) Example:

```
class Day {
  public static void main(String[] args) {

    int week = 4;
    String day;

    switch (week) {
      case 1:
        day = “Sunday”;
        break;
      case 2:
        day = “Monday”;
        break;
      case 3:
        day = “Tuesday”;
        break;
      case 4:
        day = “Wednesday”;
        break;
      case 5:
        day = “Thursday”;
        break;
      case 6:
        day = “Friday”;
        break;
      case 7:
        day = “Saturday”;
        break;
      default:
        day = “Invalid day”;
        break;
    }
    System.out.println(day);
  }
}
```

When you run the program, the output will be:

Wednesday

(b) E=Math.pow(a+b,2);

(c)

```
class PickCode
{ private char ch;
```

```
  private boolean n;
  PickCode()
  { ch= 'C';
    n = true;
   }
}
```

(d) A stream can be defined as a sequence of data. The InputStream is used to read data from a source and the OutputStream is used for writing data to a destination.

(e) Strings are immutable in Java it means once created you cannot modify content of String. If you modify it by using toLowerCase(), toUpperCase() or any other method, It always result in new String. Since String is final there is no way anyone can extend String or override any of String functionality.

(f) anytine
Java is Funamytime

(g)
```
if(n%2==0 && n>35 && n<67)
a=a+4;
if(n%2==0 && n>35 && n<67)
n=n/a;
if(n%2==0 && n>35 && n<67)
rem=n%10;
```

(h) `Pack ob=new Pack();`

(i) Constructor with parameters is called parameterised constructor, that is generally used to accept values for initialisation of its data members.

(j) It is used to ensure that multiple instances of the main() method is not created with each instances of an object.

Question 4

Define a class Book Store having the following specifications:

Data Member: Name of the book, author, publication and cost.

Member Methods:

(i) To accept the book details

(ii) To calculate the discount of 13.5% given on all books.

(iii) To display the book details.

Using a main method and an object. call the above methods

Ans.

```
import java.util.*;
class BookStore
{
  String name,author,pub;
  double cost;
  void accept()
```

```
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter book name:");
        name=sc.nextLine();
        System.out.println("Enter the name of the author:");
        author=sc.nextLine();
        System.out.println("Enter the publication:");
        pub=sc.nextLine();
        System.out.println("Enter the cost:");
        cost=sc.nextDouble();
    }
    double calc()
    {
        double dis;
        dis=13.5/100*cost;
        return dis;
    }
    void display()
    {
        System.out.println("Book name:"+name);
        System.out.println("Author:"+ author);
        System.out.println("Publication:"+pub);
        System.out.println("Dispcount:"+calc());
    }
    public static void main(String args[])
    {
        BookStore ob=new BookStore();
        ob.accept();
        ob.display();
    }
}
```

Question 5

Write a program in Java to accept a string and display the new string after removing all the vowels from the string.

Sample Input: Computer Applications with BlueJ
Sample Output: Cmptr pplctns with BlJ

Ans.

```
import java.util.*;
class Q5
{
    static void main()
    {
        String s,n=" ";
```

```
        char x;
        int i;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a string:");
        s=sc.nextLine();
        for(i=0;i<s.length();i++)
        {
           x=s.charAt(i);
           if("AEIOUaeiou".indexOf(x)==-1)
                n+=x;
        }
        System.out.println(n);
    }
}
```

Question 6

Write a program to accept numbers in a 4×4 matrix, then print the all prime numbers present in the matrix with array Index value.

SAMPLE DATA:
INPUT:
16 15 1 2
6 4 10 14
9 8 12 5
3 7 11 13
OUTPUT:
PRIME ROW INDEX COLUMN INDEX
2 0 3
3 3 0
5 2 3
7 3 1
11 3 2
13 3 3

Ans.

```
import java.util.*;
class Q6
{
   static boolean isPrime(int n)
   {
      int i,c=0;
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
            c++;
```

```
        }
        if(c==2)
            return true;
        else
            return false;
    }
    static void main()
    {
        String s,n=" ";
        char x;
        int a[][]=new int[4][4],i,j;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the matrix:");
        for(i=0;i<4;i++)
        {
            for(j=0;j<4;j++)
            {
                a[i][j]=sc.nextInt();
            }
        }
        System.out.println("PRIME\t\tROW INDEX\t\tCOLUMN INDEX");
        for(i=0;i<4;i++)
        {
            for(j=0;j<4;j++)
            {
                if(isPrime(a[i][j]))
                    System.out.println(a[i][j]+"\t\t"+i+"\t\t"+j);
            }
        }
    }
}
```

Question 7

A class Sort contains an array of 50 integers. Some of the member functions/data members are given below:

Class name Sort

Data members/instance variables

arr[] integers

item number to be searched for in the array

Member functions/methods

void inpdata() to input 50 integers.

void bubsort() to sort the array in ascending order using the bubblesort technique and to display the sorted list.

void binsearch() to input item and search for it using the binary search technique, if found to print the item searched and its position in the sorted list, otherwise to print an appropriate message.

Specify the class Sort giving the details of the functions void inpdata(), void bubsort() and void binsearch().

The main function need not be written.

Ans.

```
import java.util.*;
class Sort
{
   int arr[]=new int[50];
   void inpdata()
   {
      int i;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the value:");
      for(i=0;i<50;i++)
      {
         arr[i]=sc.nextInt();
      }
   }
   void bubsort()
   {
      int i,j,t;
      for(i=49;i>0;i--)
      {
         for(j=0;j<i;j++)
         {
            if(arr[j]>arr[j+1])
            {
               t=arr[j];
               arr[j]=arr[j+1];
               arr[j+1]=t;
            }
         }
      }
   }
   void binsearch()
   {
      Scanner sc=new Scanner(System.in);
      int n,l=0,u=49,m,f=0;
      System.out.println("Enter the number to search:");
      n=sc.nextInt();
      while(l<=u)
```

```
        {
            m=(l+u)/2;
            if(arr[m]==n)
            {
                f=1;
                break;
            }
            else if(arr[m]>n)
                u=m-1;
            else
                l=m+1;
        }
        if(f==1)
            System.out.println("Found");
        else
            System.out.println("Not found");
    }
}
```

Question 8

Write a menu driven class to accept a number and check whether it is

(a) Palprime number – [a number is a palindrome and a prime number Eg. 101]

(b) Armstrong number – [Sum of the cubes of the digits = number Eg. 153]

Ans.

```
import java.util.*;
class Q8
{

    static void main()
    {
        int n,i,ch,c=0,r=0,d;
        Scanner sc=new Scanner(System.in);
        System.out.println("MENU");
        System.out.println("1. for Pal-prime");
        System.out.println("2. for Armstrong");
        System.out.println("Enter your choice:");
        ch=sc.nextInt();
        System.out.println("Enter a number:");
        n=sc.nextInt();
        switch(ch)
```

```
    {
      case 1:
        for(i=1;i<=n;i++)
        {
          if(n%i==0)
            c++;
        }
        for(i=n;i>0;i/=10)
        {
          d=i%10;
          r=r*10+d;
        }
        if(r==n && c==2)
          System.out.println("Pal-prime");
        else
          System.out.println("Not Pal-prime");
        break;
      case 2:
        for(i=n;i>0;i/=10)
        {
          d=i%10;
          r=r+d*d*d;
        }
        if(r==n)
          System.out.println("Armstrong");
        else
          System.out.println("Not Armstrong");
      default:
        System.out.println("Invalid choice");
    }
  }
}
```

Question 9

Write a program to input a word in uppercase and rearrange the characters of the word in alphabetical order.

Ans.

```
import java.util.*;
class Q9
{
  static void main()
  {
```

```
        String w,n=" ";
        char x;
        int i;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a word:");
        w=sc.nextLine();
        w=w.toUpperCase();
        for(x='A';x<='Z';x++)
        {
           for(i=0;i<w.length();i++)
           {
               if(w.charAt(i)==x)
                 n+=x;
           }
        }
        System.out.println(n);
    }
}
```