

# Computer Applications

Solution (ICSE)

**IX**

# Chapter 1

## Principles of Object Oriented Programming



### A. Tick (✓) the correct option.

1. A set of instructions given to a computer to do a particular task.
  - a. Program
  - b. High level Language
  - c. Object
  - d. None of these

**Ans.** a. Program

2. An object is represented by two attributes, out of which one is characteristics and the other one is \_\_\_\_\_.
  - a. Behaviour
  - b. Situation
  - c. Abstraction
  - d. Encapsulation

**Ans.** a. Behaviour

3. Name the programming technique that implements programs as an organized collection of interactive objects.
  - a. Procedure Oriented Programming
  - b. Modular Programming
  - c. Object Oriented Programming
  - d. None of these

**Ans.** c. Object Oriented Programming

4. Name the programming technique that specifies a series of well-structured steps and procedures within its programming context to compose a program.
  - a. Procedure Oriented Programming
  - b. Modular Programming
  - c. Object Oriented Programming
  - d. None of these

**Ans.** a. Procedure Oriented Programming

5. 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

6. What is the behaviour aspect of an object represented by?
  - a. Member Functions
  - b. Data Members
  - c. Both a and b
  - d. None of these

**Ans.** a. Member Functions

7. What is the ability of an object to take on many forms called?
- a. Polymorphism
  - b. Encapsulation
  - c. Abstraction
  - d. Inheritance

**Ans.** a. Polymorphism

8. What is the term that is used to represent hierarchical relationship of generalization?
- a. Polymorphism
  - b. Encapsulation
  - c. Abstraction
  - d. Inheritance

**Ans.** d. Inheritance

9. Name the art of implementing Encapsulation in Object Oriented Programming.
- a. Polymorphism
  - b. Encapsulation
  - c. Abstraction
  - d. class

**Ans.** d. class

10. What is meant by state of an object?
- a. Functions of the object
  - b. Data Members of the object
  - c. Content of an object
  - d. All of these

**Ans.** c. Content of an object

**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.
- 2. Procedure Oriented Language follows top down approach.
- 3. Java is an example of Object Oriented Language.
- 4. Hiding the complexity and giving a simple interface is called Inheritance.
- 5. Abstraction is same as Encapsulation.

**C. Fill in the blanks.**

- 1. An object is an identity with certain characteristic and behaviour.
- 2. The values/attributes of the characteristics of an object are called the state of an object.
- 3. All the complexities of a program should be encapsulated in such a way so that Abstraction is obtained.
- 4. Inheritance allows us to encompass the parent class' state and behaviours into its child.
- 5. Poly-means many and Morphism means forms.
- 6. Encapsulation is a principle of Object Oriented Programming (OOP) that binds together characteristics and behaviour of an object.
- 7. Abstraction is the reduction of a particular body of data to a simplified representation of the whole.

8. The characteristics of an object are represented through data members and behaviour is represented through member functions.
9. A program written in a high level language is also called Source Code.
10. All objects have identity and are distinguishable even if the constituent components are same.

## SECTION A

### Answer the following questions.

1. **Give two examples of real world objects. Also specify their characteristics and behaviour.**

**Ans.** The *book* that you are reading now is also an example of an object. Its characteristics is represented by the information it holds, size, volume and its colour. The behavioural aspect is referred by the methods for accessing the information it contains.

A pen is also an example of object. Its characteristics is represented by its colour, shape, brand, etc., and its behaviour is represented by its use such as writing, drawing, etc.

2. **What do you understand by state of an object? Explain with an example.**

**Ans.** The *state* of an object is the particular condition it is in. For example, a lamp can be on or off. The lamp's switch (methods) turn lamp on and turn lamp off are used to access the state of the lamp.

3. **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.

4. **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.

Encapsulation is also frequently confused with abstraction, since the two concepts are closely related. Abstraction is a process of hiding the complexity and giving a simple interface. Encapsulation on the other hand is the mechanism by which the abstraction is implemented.

5. **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.

6. **Explain the term object using an example.**

**Ans.** The book that you are reading now is also an example of an object. Its characteristics is represented by the information it holds, size, volume and its colour. The behavioral aspect is referred by the methods for accessing the information it contains. For example, you can open the book, turn a page, read a paragraph, and search the table of contents, and so on. The information contained

in the book along with the methods for accessing it for gaining information is what comprises the object known as this book.

**7. 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.

**8. 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.

**9. State the four characteristics/principles of Object Oriented Programming.**

**Ans.** Encapsulation, Abstraction, Polymorphism and Inheritance.

**10. Give a real life example of Polymorphism.**

**Ans.** Example in real life of polymorphism- Shobha is a married girl and mother of 2 children doing teaching job then she is a women first, teacher in a school when she is at school, wife of someone at home, mother of her children and obvious daughter of someone. Thus you can see that Shobha plays different roles at different times that is what polymorphism is.

## Chapter 2

# Introduction to Java



### A. Tick (✓) the correct option.

1. What was Java initially called?

- a. Oak
- b. C
- c. C++
- d. None of these

**Ans.** a. Oak

2. What is Java Programming Language?

- a. A runtime system
- b. A set of development tools.
- c. An Application Programming Interface (API)
- d. All of these

**Ans.** d. All of these

3. Name the process that converts source code to bytecode.

- a. Interpretation
- b. Compilation
- c. All of these
- d. None of these

**Ans.** b. Compilation

4. A Virtual Processor that is implemented in software and runs using the capabilities provided by your operating system and computer hardware.

- a. Byte Code
- b. Compiler
- c. Interpreter
- d. Java Virtual Machine(JVM)

**Ans.** d. Java Virtual Machine(JVM)

5. Which among the following is not a language feature in Java?

- a. Robust
- b. Secured
- c. Platform Independent
- d. Procedure Oriented

**Ans.** d. Procedure Oriented

6. Name the programs that can be developed in such a way that it remains embedded in a web page and runs on the viewer's machine in a secured manner by Java compatible browsers.

- a. Applets
- b. Applications
- c. Both a and b
- d. None of these

**Ans.** a. Applets

7. Name the Application program that is written and compiled which may then be executed in any machine provided it contains the JVM.
- a. Applets
  - b. Applications
  - c. Both a and b
  - d. None of these

**Ans.** b. Applications

8. What is the extension of a source code in Java?
- a. .java
  - b. .class
  - c. Both a and b
  - d. None of these

**Ans.** a. .java

9. What is the extension of the byte code in Java?
- a. .java
  - b. .class
  - c. Both a and b
  - d. None of these

**Ans.** b. .class

10. What is a set of pseudo machine language instructions that are understood by the Java Virtual Machine and are independent of the underlying hardware called?
- a. JVM
  - b. Source Code
  - c. Compilation
  - d. Bytecode

**Ans.** d. Bytecode

**B. Fill in the blanks.**

1. James Arthur Gosling is called the father of Java programming language.
2. Java was formerly known as Oak.
3. A Java program written in a text editor is called the Source code.
4. Bytecode is a set of pseudo machine language instructions that are understood by the JVM.
5. Java is independent of the underlying hardware and software and is therefore called platform independent.
6. The process of converting a source code to byte code is called compilation.
7. A platform is the hardware/software environment in which a program remains.
8. The byte code file in Java has the .class extension.
9. JVM is a microprocessor that is implemented in software and runs using the capabilities provided by your operating system and computer hardware.
10. Multi-threading is a process of executing multiple threads simultaneously.

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

1. The byte code when compiled gives source code. F
2. A Java program can execute only in Windows machine. F
3. A Java API is a collection of prewritten packages, classes, and interfaces with their respective methods, fields and constructors. T
4. Source Code in Java has the .class extension. F
5. Java is simple to learn. T

**SECTION A**

**Short answer type questions.**

**1. Name two types of Java programs.**

**Ans.** Application and Applet

**2. What is Java API?**

**Ans.** An application programming interface (API), in the context of Java, is a collection of prewritten packages, classes, and interfaces with their respective methods, fields and constructors.

**3. Define the term Byte Code.**

**Ans.** Java bytecode is the result of the compilation of a Java program, an intermediate representation of that program which is machine independent. The Java bytecode gets processed by the Java virtual machine (JVM) instead of the processor.

**4. What is JVM? Explain its relationship with byte code.**

**Ans.** The Java Virtual Machine (JVM) is the runtime engine of the Java Platform, which allows any program written in Java or other language compiled into Java bytecode to run on any computer that has a native JVM.

**5. What is a platform? How does it differ from a Java platform?**

**Ans.** A platform is the hardware or software environment in which a program remains. The Java platform differs from most other platforms in that it's a software only platform that runs on top of other hardware-based platforms.

**6. State the difference between Applet and Application.**

**Ans.** The fundamental difference between the two Java programs is that an application program is designed to run on a stand-alone machine whereas an applet is a web-version of an application which is used to run a program on a web browser.

**7. What makes Java suitable to create secured applications?**

**Ans.** Java is designed to be secure in a networked environment. It is designed with multiple layers of security countermeasures embedded in the compiler, runtime system, and in browsers that support Java. These countermeasures make Java inherently more secure for developing any type of trusted application software. The Java run-time environment uses a byte code verification process to ensure that code loaded over the network does not violate Java security constraints.



**8. What does the Robust feature in Java mean?**

**Ans.** The term robust means strong or vigorous. Java uses some strong design measures to eliminate certain types of programming errors. It allows extensive compile-time error checking. Therefore most of the errors will be trapped and need to be rectified during compilation.

**9. State four features of Java.**

**Ans.** Simple, Robust, Secured and Platform Independent.

**10. What is Multithreading in Java?**

**Ans.** Multithreading is a process of executing multiple threads simultaneously. Thread is basically a lightweight sub-process, a smallest unit of processing. The main purpose of multithreading in Java is to provide simultaneous execution of two or more parts of a program to utilize the CPU to the maximum.

**11. What makes Java Architecture Neutral?**

**Ans.** In Java there are no implementation dependent features. In most of other languages you need to know features that are hardware or OS dependent. For example in a high-level language called C the int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.

**12. What is WORA?**

**Ans.** “Write once, run anywhere” (WORA), or sometimes write once, run everywhere (WORE), is a slogan created by Sun Microsystems to illustrate the cross-platform benefits of the Java language.

**13. Explain the statement “Byte code is the object code for the JVM”.**

**Ans.** Normally object code refers to a program in a form that the host system can run directly. As java source code is converted to bytecode during compilation which is then recognised by the JVM, Thus bytecode is sometimes referred to as the object code for the JVM.

# Chapter 3

## Elementary Concept of Objects and Classes



### A. Tick (✓) the correct option.

1. Which among the following creates a blueprint to represent characteristic and behaviour of an object?
- a. Object
  - b. Class
  - c. Instance
  - d. None of these

**Ans. b. Class**

2. Which among the following do not belong to an object?
- a. State
  - b. Behaviour
  - c. Identity
  - d. Class

**Ans. a. State**

3. Which among the following is not a component of a class?
- a. Modifiers
  - b. Class name
  - c. Object
  - d. Body

**Ans. c. Object**

4. Which among the following keyword is used to allocate memory space for an object?
- a. new
  - b. for
  - c. while
  - d. int

**Ans. a. new**

5. 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)**

6. 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**

7. Which among the following is a valid class name?
- a. Simple Interest
  - b. SimpleInterest
  - c. 1SimpleInterest
  - d. Simple@Interest

**Ans. b. SimpleInterest**

8. Which among the following is a valid object name?
- a. obj1
  - b. 1obj
  - c. Obj 1
  - d. Obj#1

**Ans. a. obj1**

9. Which among the following is used to represent behaviour in a class?
- a. Method
  - b. Data members
  - c. Both a and b
  - d. None of these

**Ans. a. Method**

10. If a method named show( ) is to be invoked using an object 'ob' , which among the following is correct?
- a. ob.show( )
  - b. ob.show
  - c. show( ).ob
  - d. None of these

**Ans. a. ob.show()**

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

- 1. An object is called a class factory. F
- 2. A class is an instance of an object. F
- 3. A class is a mechanism to implement encapsulation. T
- 4. Data members in a class is used to represent the characteristic of an object. T
- 5. The new operator is used to create an object. T
- 6. It's a rule to have a class-name beginning in capital letter. F
- 7. Java is case sensitive. T
- 8. A class-name cannot be a keyword. T
- 9. The dot operator is used to access members in an object using an object. T
- 10. In programming every object will have a name. T

**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. It is a common convention to begin a class-name in **capital** letter.
7. A class is called an **object** factory.
8. **Object-name** is used to give a unique name to an object for identification.
9. Behaviour of an object is represented by **methods**.
10. The size of fundamental data type is **fixed**.

## SECTION A

Answer the following questions.

1. State two differences between fundamental and user-defined data type.

Ans.

Comparison of Fundamental & User defined data type.	
Fundamental data type	User defined 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.

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

Ans.

A class	An object
Class is a <b>blueprint or template</b> from which objects are created.	Object is an <b>instance</b> of a class.
Class is a <b>group of similar objects</b> .	Object is a <b>real world entity</b> such as pen, laptop, mobile, bed, keyboard, mouse, chair etc.
Class is a <b>logical</b> entity.	Object is a <b>physical</b> entity.

3. Why is a class called a user-defined data type?

Ans. Primitive **data types** are the general and fundamental **data types** that we have in **Java** and those are byte, short, int, long, float, double, char, boolean, etc., **User defined data types** are those that **user / programmer** himself defines.

**4. What is an access specifier? Which two access specifier is used in a class declaration?**

**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.

The public and default access specifier is used in a class declaration.

**5. 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.

The following statements create two objects s1 and s2 of the class Student.

```
Student s1 = new Student();
```

```
Student s2 = new Student();
```

So, we have a single class Student but we can create as many objects as we want (like s1, s2, etc.) from that single class.

This is similar to what happens in a factory. Consider a factory which produces car. They have only a single design of a car but produce multiple cars from that single design.

Things are similar in the world of classes and objects. There is a single definition of a particular class (like Student) but we can produce many Student objects (like s1, s2) from that single class.

**6. State two rules you should follow for naming a class.**

**Ans.** While using names for a class the following set of rules are to be kept in mind.

1. It can have any alphabet (capital or small), digits, underscore and dollar sign characters. For example, *a, b, cat, mat123, cost\_price, Topaz\$* are all example of valid identifier.
2. It should not begin with digits or should not contain any special character. For example *2ab, ab#c, top@, etc.*, are invalid identifiers as it either begins with a digit or contain special characters (like #, @).
3. It cannot have a space between it. For example, *Simple Interest* or *Selling Price* are invalid class-names as it contains a space.
4. It must not be a *keyword*. For example, *for, while, do* are invalid class-names as they are keywords and are assigned some special function for the language.
5. It can be of any length. Even though Java gives you the flexibility to provide a huge length for an identifier a very long name is impractical and difficult to manage.

**7. What two conventions you should follow naming a class?**

**Ans.** Conventions for class name:

1. Class name should begin with capital letter. Example, *Bank, School, Student*
2. If a class name consists of more than one word, the first letter of each new word should be in uppercase. For example *MyClass, AccountDetails, SimpleInterest*, etc.

8. State two conventions you should follow while naming a class.

**Ans.** Class name should begin with capital letter. Example, Bank, School, Student

If a class name consists of more than one word, the first letter of each new word should be in uppercase. For example MyClass, AccountDetails, SimpleInterest, etc.

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

**Ans.** SimpleInterest si=new SimpleInterest( );

10. Rewrite the following program after removing the errors, underlining each corrections:

```
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( );
    }
}

```

**11. Which among the following are invalid class name in Java? State with reasons.**

**1. Compound Interest**

**Ans.** Invalid as it contains a space.

**2. 1MyClass**

**Ans.** Invalid as it should not begin with a digit.

**3. MyClass\$**

**Ans.** Valid

**4. Myclass#**

**Ans.** Invalid as it contains a special character #.

**5. My@Class**

**Ans.** Invalid as it contains a special character @.

# Chapter 4

## Values and Data Types



### A. Tick (✓) the correct option.

1. Which among the following encoding system can represent 65536 characters?
- a. ASCII
  - b. Unicode
  - c. Both a and b
  - d. None of these

**Ans.** b. Unicode

2. Which among the following is not a Token?
- a. Keywords
  - b. Literals
  - c. Identifiers
  - d. Data Type

**Ans.** d. Data Type

3. Which among the following is a keyword but not a literal?
- a. for
  - b. null
  - c. true
  - d. false

**Ans.** a. for

4. 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

5. Which among the following is a valid octal integer literal?
- a. 0178
  - b. 675
  - c. 0675
  - d. 0X675

**Ans.** c. 0675

6. 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;

7. Which among the following is not a punctuator?
- a. ; semicolon
  - b. , comma
  - c. : colon
  - d. . dot

**Ans.** c. : colon



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

- a. int
- b. float
- c. String
- d. char

**Ans.** c. String

9. 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

10. 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. Fill in the blanks.

1. **Character Set** is a valid set of characters used in a Java language.
2. A **unicode** is a type encoding that can represent characters of all the languages of the world.
3. **Token** is the fundamental building block of a program.
4. **Keywords** are reserved words in Java that provide a special meaning to the java compiler.
5. The names given to different parts of a program are called **Identifiers**.
6. **Octal** integer literals are always prefixed with 0.
7. Literals that are enclosed within double quotes are called **String** literals.
8. Symbolic constant declarations are always preceded with the **final** keyword.
9. **Operators** are special symbols that perform specific operations on one or more operands and then return a result.
10. The true and false literals are both **boolean** literals.

## SECTION A

### A. Very short answer type questions

1. Identify the mistake in each of these lines and correct it:

a. `int goto=5;`

**Ans.** The identifier goto is invalid, as it is an identifier.

Correction: `int g=5;`

b. `float a=12.3;`

**Ans.** The float literals should be suffixed with 'f'.

Correction: `float a=12.3f;`

**c. doubles b=15;**

**Ans.** The data type is wrongly spelled.

Correction: double b=15;

**d. long b=1536458888397632;**

**Ans.** The long literal should be suffixed with an 'L'.

**Correction: long b=1536458888397632L;**

e. String s= 'Computer';

**Ans.** String literals should be enclosed in double quotes.

**Correction: String s= "Computer";**

**2. State with reasons why are the following variable names invalid:**

**a. for**

**Ans.** Invalid, as for is a keyword.

**b. book list**

**Ans.** Invalid, as a space cannot be present in a variable name.

**c. 2ndTerm**

**Ans.** Invalid, as a variable name cannot begin with a digit.

**d. hash#**

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

**e. sick@always**

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

**3. 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**

**Ans.** float

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

**Ans.** float

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

**Ans.** double

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

**Ans.** int.

**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.** double

## B. Short answer type questions

State with reasons why are the following initializations incorrect:

a. `int a=5;`  
`short b=a;`

**Ans.** The `short b=a;` is invalid as an `int` cannot be initialised to a short type variable.

b. `double a=5.3;`  
`float b=a;`

**Ans.** The `float b=a;` is invalid as a which is of double type cannot be initialised to a variable of float type.

c. `int a=01238;`

**Ans.** The `int a =01238;` is invalid as an starting with a 0 is considered to be of octal data type and therefore cannot have a digit 8.

d. `float a=17.36f;`  
`int b=a;`

**Ans.** The `int b=a;` is invalid as a is of float type, that cannot be initialised to a variable b of int type.

e. `boolean a=true;`  
`int b=a;`

**Ans.** The `int b=a;` is invalid as the assignment is of incompatible type.

## C. Answer the following questions.

1. What is Character Set?

**Ans.** A character set is a set of alphabets, letters and some special characters that are valid in Java language.

2. What are keywords?

**Ans.** A keyword is a reserved word that have a special significance to the compiler and cannot be used anywhere else other than what it is intended for.

3. What are Identifiers? State the rules while using Identifiers.

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

While using identifiers the following set of rules are to be kept in mind.

1. It can have any alphabet (capital or small), digits, underscore and dollar sign characters. For example, *a, b, cat, mat123, cost\_price, Topaz\$* are all example of valid identifier.
2. It should not begin with digits or should not contain any special character. For example *2ab, ab#c, top@,* etc, are invalid identifiers as it either begins with a digit or contain special characters (like #, @).
3. It cannot have a space between it. For example, simple interest or selling price are invalid identifiers as it contains a space.

4. It must not be a keyword. For example, for, while, do are invalid identifiers as they are keywords and are assigned some special function for the language.
5. It can be of any length. Even though Java gives you the flexibility to provide a huge length for an identifier a very long name is impractical and difficult to manage.

**4. What is a literal? What are the different types of literals available in Java?**

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

The different types of literals in Java are:

1. Integer-literal or Fixed point-literal
2. Floating point-literal
3. Boolean-literal
4. Character-literal
5. String-literal
6. Null-literal
7. Class literal

**5. State the difference between a boolean literal and a character literal.**

**Ans.** A boolean literal may be either true or false and character literal are always enclosed within single quotes.

**6. How are Decimal, Octal and Hexadecimal integer literals represented in Java?**

**Ans.** Decimal Integer Literals uses digits from 0 to 9 (i.e. 10 digits this is what you normally use to represent numbers in mathematics) but should not be prefixed with 0. + (or no symbol) or – may be used to represent positive or negative numbers respectively. Thus -12, 56, or +75 may be considered to be Decimal Integer Literal.

Octal Integer Literals uses digits from 0 to 7 (i.e. 8 digits) and should be prefixed with 0. + (or no symbol) or – may be used to represent positive or negative numbers respectively. Thus -012, 056, or +075 may be considered to be Octal Integer Literal.

Hexadecimal Integer Literals uses digits from 0 to 9 and alphabets from A (or a) to F (or f) representing 10 to 15 (i.e. 16 numbers) and should be prefixed with 0x or 0X. + (or no symbol) used to represent positive numbers or – may be used to represent negative numbers. Thus -0x12, 0X56, or +0A7F5 may be considered to be Hexadecimal Integer Literal.

**7. Write any two escape sequences used in Java.**

**Ans.** For newline :\n

For tab: \t

**8. What are the different punctuators available in Java?**

**Ans.** There are 9 punctuators available in Java. They are:

(){}[];,. .

**9. What are Tokens?**

**Ans.** Token is the fundamental building block of a program that represents a single element of a programming language.

**10. State the difference between token and identifier.**

**Ans.** Token is the fundamental building block of a program that represents a single element of a programming language. Identifier on the other hand is a token that is used to name different parts of a program.

**11. State the two kinds of data types.**

**Ans.** Primitive data type and Composite data type.

**12. What is the smallest and the largest number that can be stored into a variable of int data type?**

**Ans.** Smallest:  $-2^{31}$   
Largest =  $2^{31} - 1$

**13. How are floating point numbers represented in Java?**

**Ans.** It should be suffixed with f or F.

**14. What are variables? How are variables initialized?**

**Ans.** Variables are names given to memory location. Variables are initialised during declaration. For example,

```
int a=25;
```

**15. How are symbolic constants represented in Java?**

**Ans.** Symbolic constants are prefixed with the final keyword during declaration. For example,  
final int a =100;

**16. What are separators?**

**Ans.** Separators are special symbols used to separate statements, data elements or mark blocks in Java and is therefore also called separators. There are 9 punctuators available in Java. They are:  
( ) { } [ ] ; , .

**17. What is Type Conversion? Name the two type conversions in Java.**

**Ans.** In a mathematical expression containing operands of more than one data type necessitates to know the resultant data type of an expression. The resultant data type of an expression can be categorized into two segments.

- Implicit type conversion
- Explicit type conversion

**18. Name two keywords which are also literals in Java.**

**Ans.** They are null, true and false.

**19. What do you understand by variable scope?**

**Ans.** Variable scope is the area in which a variable is accessible or have its life.

**20. What are the three types of integer literals that can be represented in Java?**

**Ans.** Decimal Integer Literal, Octal Integer Literal and Hexadecimal Integer Literal.

# Chapter 5

## Operators in Java



Tick (✓) the correct option.

1. Which among the following is a relational operator?

- a. +                      b. >                      c. \*                      d. /

**Ans.** a. +

2. The && operator requires which among the following conditions to be satisfied to represent a success for two relations?

- a. Both should be satisfied                      b. Any one should be satisfied.  
c. None of them are satisfied.                      d. None of these

**Ans.** a. Both should be satisfied

3. Which among the following is not a logical operator?

- a. &&                      b. ||                      c. !                      d. <=

**Ans.** d. <=

4. Which among the following operators is used to dynamically allocate memory space for an object?

- a. . (dot)                      b. >                      c. ==                      d. new

**Ans.** d. new

5. Which among the following is the correct construct for creating an object named 'book' of the 'Library' class?

- a. Library book=new Library;                      b. book =new Library();  
c. Library() book=new Library();                      d. Library book=new Library();

**Ans.** d. Library book=new Library();

6. Which operator is used to access individual members of an object?

- a. . (dot)                      b. >                      c. ==                      d. New

**Ans.** a. . (dot)

7. The member 'out' object of the 'System' class belongs to which class?

- a. System                      b. Print                      c. PrintStream                      d. Println

**Ans.** a. System

8. The statement `a%2` will result to what if the value of `a` is an even number?  
a. 0                      b. 1                      c. Both a and b                      d. None of these

**Ans.** a. 0

9. Which among the following operators is used to check the two operands are equal or not?  
a. Equals                      b. =                      c. ==                      d. None of these

**Ans.** c. ==

10. Which among the following forms of operators works with three operands?  
a. Unary                      b. Binary                      c. Ternary                      d. None of these

**Ans.** c. Ternary

## SECTION A

### Answer the following questions

1. **What is an operator? How is the classification of operators based upon?**

**Ans.** An operator is a specialised symbol that performs a specific kind of evaluation on one, two, or three operands, and produces a result.

Operators in Java can be categorized based on two criteria:

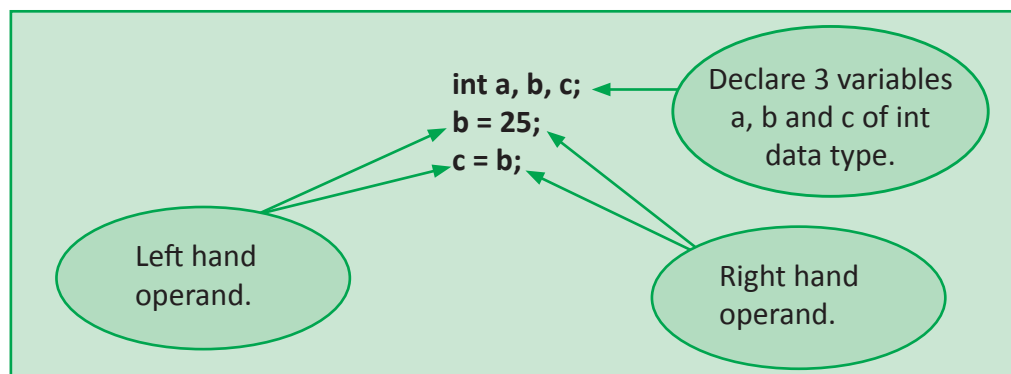
- Forms of operators
- Types of operators

2. **What are the different forms of operators?**

**Ans.** Forms of operators groups operators in terms of the number of operands. There are three types of operators based on the number of operands. An operator is called a unary, binary, or ternary operator based on the number of operands. If an operator takes one operand, it called a unary operator; if it takes two operands, it called a binary operator; if it takes three operands, it called a ternary operator.

3. **What is assignment operator? Explain with the help of an example.**

**Ans.** The Assignment operator is used to assign or give a value to a variable.



**4. What are Arithmetic operators? What are the different forms of arithmetic operators?**

**Ans.** Arithmetic operators are used to evaluate mathematical expressions.

Arithmetic operators are a form of operator that may either be a Unary Operator or a Binary Operator.

**5. What is increment/decrement operator used for? Explain with the help of an example.**

**Ans** The increment/decrement operator is used to increase/decrease the value of a variable by 1.

For example `a++`; increases the value of the variable `a` by 1, similarly `a--`; decreases the value of the variable `a` by 1.

**6. What is an expression?**

**Ans.** An expression is a construct that is made up of variables, operators and method invocations, which are constructed according to the syntax of the language that evaluates to a single value.

**7. State the different types of binary arithmetic operators. Also state their hierarchy of operations.**

**Ans** These are operators that work on two operands. The operator is placed between the operands. The symbols that are used as binary arithmetic operators are:

- + For performing addition
- For performing subtraction.
- \* For performing multiplication.
- / For performing division.
- % For getting the remainder when divided.

**Hierarchy of Arithmetic Operators**

1. First Brackets/Parenthesis: In Java the expression in bracket is done first. However brackets in an expression should only be first bracket ( ). Nested brackets can also be used, but in that case the expressions in the innermost bracket are done first and then the subsequent outer-level brackets. Eg.  $(5+6*(7-3))$  will give 29. Notice that only first-bracket is used.
2. Then /, \*, or % whichever comes first while evaluating from left to right will get evaluated first.
3. Then + or – whichever comes first while evaluating from left to right will get evaluated first..

**8. What is Operator Associativity? What is the operator associativity of prefix/postfix increment/decrement operators?**

**Ans.** Operator Associativity 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.

Operator	Description	Associativity
<code>++</code>	Post-Increment	left to right
<code>--</code>	Post-Increment	
<code>++</code>	Pre-Increment	right to left
<code>--</code>	Pre-Increment	



**9. What are short hand operators? Explain with an example.**

**Ans.** Java provides special operators that can be used to combine an arithmetic operation with an assignment. For example, the expression,

**a = a+4;**

can be written as,

**a += 4;**

As this operator is a shorter way of expressing an expression it is also sometimes called Short hand operators.

**10. What is an accumulator?**

**Ans.** An accumulator is a variable that the program uses to calculate a sum or product of a series of values.

**11. What are counters?**

**Ans.** Counter is a variable that keeps track of the number of times an operation has been performed.

**12. Name the relational operators along with their usage in Java.**

**Ans.**

Relational operator	Usage
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
= =	Equal to
!=	Not equal to

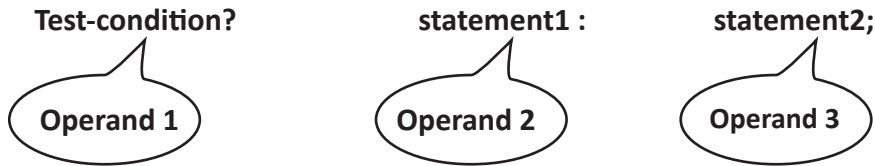
**13. Name the logical operators used in Java.**

**Ans.**

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.

14. Give the general syntax of conditional operator and also label its operands.

Ans. It has the following general syntax:



15. What is the use of:

- a. Dot (.) operator
- b. new operator

Ans. a. The dot operator is used to access or invoke individual members of a class.  
b. The new operator dynamically allocates memory space for an object.

**B. What will be the output of the following program?**

```
1. 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);
}
}
```

Ans. 5  
15

```
2. 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);
}
```

```
    }  
}
```

**Ans.**

19  
-37

3. **class Output3**

```
{  
    static void main()  
    {  
        int a=12,b=13,c=0,d=1;  
        c+=a;  
        c+=b;  
        d*=a;  
        d*=b;  
        System.out.println(c);  
        System.out.println(d);  
    }  
}
```

**Ans.** 25

156

4. **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);  
    }  
}
```

**Ans.**

14	14	27
16	15	35

5. class Output5

```
{
    static void main()
    {
        int a=12,b=13;
        boolean c;
        c=a++ > b++;
        System.out.println(a+"\t"+b+"\t"+c);
        c=++a < --b;
        System.out.println(a+"\t"+b+"\t"+c);
    }
}
```

Ans.

13	14	false
14	13	false

C. Short Answer Type Questions.

1. Write the corresponding expressions for the following mathematical operations:-

i.  $c = a^2 + b^2$

ii.  $z = x^3 + y^3 - xy/z$

Ans. i.  $c = a*a + b*b$

ii.  $z = x*x*x + y*y*y - x*y/z$ ;

2. What will be the output for the following program segment?

```
int a = 0, b = 30, c = 40;
```

```
a = --b + c++ + b;
```

```
System.out.println("a =" + a);
```

Ans. a=98

3. if  $m = 5$  and  $n = 2$  output the values of  $m$  and  $n$  after execution in (i) and (ii):-

i.  $m -= n$ ;

ii.  $n = m + m/n$ ;

Ans. i.  $m=3$  and  $n=2$

ii.  $m=5$  and  $n=7$

4. What will be the output of the following, if x = 5 initially?

i.  $5 * ++x$

ii.  $5 * x++$

**Ans.** i. 30

ii. 25

5. Evaluate the following expressions, if the values of the variables are a = 2, b=3 and c=9

i.  $a - (b++) * (-c)$

ii.  $a * (++b) \% c$

**Ans.** i. - 22

ii. 8

6. If a = 5, b = 9 calculate the value of:

$a+ = a++ - ++b + a$

**Ans.** a=6 and b=10

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

`int x = 5;`

`x = x++ * 2 + 3 * -x;`

**Ans.** x=-8

8. State the value a, b and c after the execution of each of the following statements where a=12, b=13 and c=11:

i.  $a=a++ + -b + c++;$

**Ans.** a=35, b= 12 and c= 12

ii.  $b=b++ + ++a * 2;$

**Ans.** a=13, b= 39 and c=11

iii.  $c=c++ + a++ * (++b);$

**Ans.** a=13, b= 14 and c=179

iv.  $b=b++ + ++b + c++ + ++a;$

**Ans.** a=13, b= 52 and c= 12

v.  $b+= -c + ++c + a++ + (b++);$

**Ans.** a=13, b= 59 and c=11

9. Write the following statements using correct usage of the operators:

i. X is greater than equal to Y.

**Ans.**  $X \geq Y$

ii. X is not equal to Y.

**Ans.**  $X \neq Y$

iii. X is equal to Y.

**Ans.**  $X == Y$

iv. X is less than Y and X is less than Z.

**Ans.**  $X < Y \ \&\& \ X < Z$

v. Either X is greater than Y or X is less than Z.

**Ans.**  $X > Y \ || \ X < Z$

vi. If X has a value greater than 20 then increment the value of J by 1 otherwise decrement the value by 2.

**Ans.**  $J = (X > 20) ? J + 1 : J - 2;$

## 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,p;
```

```
s=a+b;
```

```
d=a-b;
```

```
p=s*d;
```

```
System.out.println("Product="+p);
```

```
}
```

```
}
```

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 d;
```

```
d=(float)(a+b+c)/3;
```

```
System.out.println("Average="+d);
```

```
}
```

```
}
```

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;
```

```
d=100-(a+b+c);
```

```

        System.out.println("Answer="+d);
    }
}

```

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\nMango\nGuava\nPomegranate\nJack fruit");
    }
}

```

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(s+"\t"+d+"\t"+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 using variables to find the cost of 17 pencils if the cost of one pencil=₹ 2.50.

**Output should be:**

**Cost of 1 pencil=₹2.5**

**Cost of 17 pencils=₹42.5**

**Ans.** class Q11

```

{
static void main()

```

```

    {
    float cp=2.50f,n=17,tc;
    tc=cp*n;
    System.out.println("Cost of 1 pencil=Rs"+cp);
    System.out.println("Cost of 17 pencils=Rs"+tc);
    }
}

```

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=  $\pi r^2$**

**Circumference of a circle =  $2 \pi r$**

**Where r is the radius of the circle and  $\pi$  is 3.142.**

**Ans.** class Q12

```

{
static void main()
    {
    float r=12,pi=3.142f,a,c;
    a=pi*r*r;
    c=2*pi*r;
    System.out.println("Areas"+a);
    System.out.println("Circumference"+c);
    }
}

```

13. Write a program to initialize two integer variables a and b 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,t;
    t=a;
    a=b;
    b=t;
    }
}

```

```
        System.out.println(a+"\t"+b);
    }
}
```

- 14. Write a program to initialize 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,l1,l2,l3,d;
    d=a%10+b%10+c%10;
    System.out.println("Sum="+d);
}
}
```

- 15. Write a program to initialize 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 e and display both these digits.**

**Ans.** class Q15

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

- 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 6

# Input in Java



Tick (✓) the correct option.

1. Which among the following is not 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. Initialisation
  - b. Parameterised input
  - c. Scanner input
  - d. None of these

**Ans.** c. Scanner input

## SECTION A

**Answer the following questions.**

- 1. In programming what is considered to be an Input?**

**Ans.** Programming involves taking in some data, processing it using some instructions and giving some meaningful result. Taking in some data is called Input.

- 2. What is initialization of variables? Explain with a help of an example.**

**Ans.** Initialization is the assignment of an initial value for a variable during declaration.

For example, `int a=5;` is a statement where 5 is stored initially into a memory location represented with the name a.

- 3. What do you understand by parameterized input?**

**Ans.** The process of passing values to arguments in a method during execution is called parameterised input.

- 4. What is the function of a Scanner class? In which package does it belong?**

**Ans.** The scanner class is a class in `java.util`, which allows the user to read values of various types, from either keyboard or a file.

- 5. Name the functions of the Scanner that is used to:**

**a. Accept a number of long data type**

**b. Accept a group of alphabets.**

- Ans.** a. nextLong( )  
b. nextLine()

**6. What is a bug? What is debugging?**

**Ans.** A bug is an error, flaw, failure or fault in a program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

The process of removing a bug is called debugging.

**7. What are the different type of errors that may occur in a Java program?**

**Ans.** Errors in a program in Java can broadly be classified into three categories:

- Syntax errors
- Logical errors
- Run-time errors

**8. 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.

The common syntax errors are:

- Missing semicolon
- Mismatched braces in classes and methods.

**9. What are run-time errors? Give two examples.**

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

The run-time errors are:

- Dividing an integer by zero.
- Accessing an element that is out of bounds of an array.

**10. 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. The different types of comments in Java are:

- Singleline comment
- Multiline comment

## Removing Errors

**1. Rewrite the following program after removing the errors in the program, if any:**

```
class Myclass
{
    import java.util.Scanner;
```

```

    static void function()
    {
        int a;
        a=sc.nextInt();
        Scanner sc=new Scanner(System.in);
        System.out.println(a);
    }
}

```

**Ans.**

```

import java.util.Scanner;
class Myclass
{

    static void function()
    {
        int a;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter an integer:");
        a=sc.nextInt();
        System.out.println(a);
    }
}

```

2. Rewrite the following program after removing the errors in the program, if any:

```

class Myclass
{
    static void main()
    {
        int a=5,b=6;
        c=a+b;
        System.out.println("Sum="a);
    }
}

```

**Ans.**

```

class Myclass
{

```

```

static void main()
{
    int a=5,b=6, c;
    c=a+b;
    System.out.println("Sum="+c);
}
}

```

3. Rewrite the following program after removing the errors in the program, if any:

```

class Myclass
{
    static void main(int a, int b)
    {
        int c;
        a+b=c;
        System.out.println("Sum="a);
    }
}

```

**Ans.**

```

class Myclass
{
    static void main(int a, int b)
    {
        int c;
        c=a+b;
        System.out.println("Sum="+c);
    }
}

```

4. Rewrite the following program after removing the errors in the program, if any:

```

class Myclass
{
    static void break(int a, int b)
    {
        int c;          /* Finding the sum of

```



```

        c=a+b;           2 numbers*/
        System.out.println("Sum="+c);
    }
}

```

**Ans.**

```

class Myclass
{
    static void sum(int a, int b)
    {
        int c;           //Finding the sum of 2 numbers
        c=a+b;           System.out.println("Sum="+c);
    }
}

```

### Questions on Output

1. Give the output of the following program:

```

public class Output
{
    static void function()
    {
        int a=3,b=14;
        System.out.println(a+ " " +b);
        a=a+b;
        b=a-b;
        a=a-b;
        System.out.println(a+" "+b);
    }
}

```

**Ans.**

```

3 14
14 3

```

2. Give the output of the program:

```

public class Output
{

```

```

static void input(int a,int b)
{
    int c,d;
    c=(++a) + (b++);
    d=(a--)- (--b);
    System.out.println(c+" "+d);
}
}

```

When:

- a. a=12 and b=13 is taken as parameters.
- b. a=9 and b=4 is taken as parameters.

**Ans.**

- a. 26 0
- b. 14 6

3. Give the output of the program:

```

public class Output
{
    static void main()
    {
        int a=5,b=6,c=0;
        c=(a+b>8)?2*a:3*b;
        System.out.println(c);
    }
}

```

**Ans.** 10

What will be the change in output if a and b is initialized to 2 and 3 respectively?

9

4. Give the output of the program:

```

public class Output
{
    static void main()
    {
        int a=5,b=6,c=3;

```

```

        c+=a/2+b%8+c;
        b-=a-b*2+c*3;
        System.out.println(c);
    }
}

```

**Ans.**

14

5. Give the output of the program:

```

public class Output
{
    static void main(int a,int b)
    {
        int s=0;
        s+=(a>0)?a:-b;
        s-=(a<0)?-a:+b;
        System.out.println(s);
    }
}

```

When

a. a=-5 and b=-6

b. a=5 and b=6

**Ans.**

a. 1

b. -1

## SECTION B

### Programs:

1. Write a program to input two integers using parameters and find their sum and product.

**Ans.**

```

class Q1
{
    void main(int a, int b)
    {
        int c,d;

```

```

        c=a+b;
        d=a*b;
        System.out.println("Sum="+c);
        System.out.println("Product="+d);
    }
}

```

**2. Write a program to input five integers and find their average.**

**Ans.**

```

import java.util.*;
class Q2
{
static void main()
    {
        Scanner sc=new Scanner(System.in);
        int a,b,c,d,e;
        float av;
        System.out.println("Enter 5 integers:");
        a=sc.nextInt();
        b=sc.nextInt();
        c=sc.nextInt();
        d=sc.nextInt();
        e=sc.nextInt();
        av=(float)(a+b+c+d+e)/5;
        System.out.println("Average="+av);
    }
}

```

**3. Write a program to input two integers using Scanner and find the product of their sum and difference.**

**Ans.**

```

import java.util.*;
class Q3
{
static void main()

```

```

{
Scanner sc=new Scanner(System.in);
int a,b,c,d,e;
System.out.println("Enter 2 integers:");
a=sc.nextInt();
b=sc.nextInt();
c=a+b;
d=a-b;
e=c*d;
System.out.println("Product="+e);
}
}

```

4. Write a program to input three integers and find the difference between their sum and their average.

**Ans.**

```

import java.util.*;
class Q4
{
static void main()
{
Scanner sc=new Scanner(System.in);
int a,b,c,s;
float av,d;
System.out.println("Enter 3 integers:");
a=sc.nextInt();
b=sc.nextInt();
c=sc.nextInt();
s=a+b+c;
av=s/3f;
d=s-av;
System.out.println("Difference="+d);
}
}

```

5. Write a program to input the Principal, Rate and Time for a certain amount of money and print the Simple Interest.

**Note:**

**Ans.**

```
import java.util.*;
class Q5
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        float p,r,t,si;
        System.out.println("Enter the principal:");
        p=sc.nextFloat();
        System.out.println("Enter the rate:");
        r=sc.nextFloat();
        System.out.println("Enter the time:");
        t=sc.nextFloat();
        si=(p*r*t)/100;
        System.out.println("Simple Interest="+si);
    }
}
```

6. Write a program to input the length and breadth of a rectangle and find its area and perimeter.

**Note: Area of a rectangle=length\*breadth**

**Perimeter of a rectangle=2\*(length+breadth)**

**Ans.**

```
import java.util.*;
class Q6
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        float l,b,a,p;
        System.out.println("Enter the length:");
```

```

l=sc.nextFloat();
System.out.println("Enter the breadth:");
b=sc.nextFloat();
a=l*b;
p=2*(l+b);
System.out.println("Area="+a);
System.out.println("Perimeter="+p);
}
}

```

7. Write a program to input the radius of a circle and find its area and perimeter.

**Note:** Area of a circle= $22/7 * \text{radius}^2$

**Circumference of a circle=  $2 * 22/7 * \text{radius}$**

**Ans.**

```

import java.util.*;
class Q7
{
static void main()
{
Scanner sc=new Scanner(System.in);
float r,a,pi=3.142f,c;
System.out.println("Enter the radius:");
r=sc.nextFloat();
a=pi*r*r;
c=2*pi*r;
System.out.println("Area="+a);
System.out.println("Circumference="+c);
}
}

```

8. Write a program to input the length, breadth and height of a cuboid and find its Volume and Total Surface Area.

**Note:** Volume of a cuboid= length\*breadth\*height

**Total Surface Area=2\*(length\*breadth+breadth\*height+height\*length)**

**Ans.**

```
import java.util.*;
class Q8
{
static void main()
    {
Scanner sc=new Scanner(System.in);
float l,b,h,v,a;
System.out.println("Enter the length:");
l=sc.nextFloat();
System.out.println("Enter the breadth:");
b=sc.nextFloat();
System.out.println("Enter the height:");
h=sc.nextFloat();
v=l*b*h;
a=2*(l*b+b*h+h*l);
System.out.println("Volume="+v);
System.out.println("Area="+a);
    }
}
```

9. Write a program to input the radius and height of a cylinder and find its volume and total surface area.

**Note:** Total Surface Area= $2\pi r^2 + 2\pi rh$

Volume= $\pi r^2 h$

**Ans.**

```
import java.util.*;
class Q9
{
static void main()
    {
Scanner sc=new Scanner(System.in);
float r,h,v,a,pi=3.142f;
```



```

System.out.println( "Enter the radius:");
r=sc.nextFloat();
System.out.println( "Enter the height:");
h=sc.nextFloat();
v=pi*r*r*h;
a=2*pi*r*r+2*pi*r*h;
System.out.println( " Volume="+v);
System.out.println( "Area="+a);
}
}

```

**10. Write a program to input three integers and find the sum of the last digit of the numbers.**

**For example if the inputs are: 12**

**26**

**35**

**Output:**

**Sum of the last digit of the integers are: 13**

**Ans.**

```

import java.util.*;
class Q10
{
static void main()
{
Scanner sc=new Scanner(System.in);
int a,b,c,s;
System.out.println("Enter the 3 integers:");
a=sc.nextInt();
b=sc.nextInt();
c=sc.nextInt();
s=a%10+b%10+c%10;
System.out.println( "Sum of the last digits="+s);
}
}

```

11. Write a program to input the temperature in Fahrenheit and change it to Celsius.

**Note:** The relation between Fahrenheit and Celsius is given by the formula:

Where C=Celsius and F=Fahrenheit

**Ans.**

```
import java.util.*;
class Q11
{
static void main()
    {
Scanner sc=new Scanner(System.in);
float f,c;
System.out.println( "Enter temperature in fahrenheit:");
f=sc.nextFloat();
c=((f-32)*5)/9;
System.out.println("Temperature in celcius="+c);
    }
}
```

12. Write a program to input time in seconds and display the time broken up as hours, minutes and seconds.

**For Example:**

**INPUT: Enter Time in Seconds: 4326**

**OUTPUT:**

**Time in hours:1**

**Time in minutes:12**

**Time in seconds:6**

**Ans.**

```
import java.util.*;
class Q12
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int sec,h,m,s;
```

```

System.out.println( "Enter time in seconds:");
sec=sc.nextInt();
h=sec/3600; //finding hours
sec=sec%3600; //remaining seconds
m=sec/60; //finding minutes
s=sec%60; //remaining seconds
System.out.println( "Time in hours="+h);
System.out.println( " Time in minutes="+m);
System.out.println( " Time in seconds="+s);
}
}

```

13. Write a program to input a floating point number and round it off to the nearest integer.

**For Example:**

**INPUT: Enter a floating point number: 12.3**

**OUTPUT: Rounded off to nearest integer:12**

**INPUT: Enter a floating point number: 14.5**

**OUTPUT: Rounded off to two places of decimal:15**

**Ans.**

```

import java.util.*;
class Q13
{
static void main()
{
Scanner sc=new Scanner(System.in);
float n;
int r;
System.out.println( " Enter a floating point number:");
n=sc.nextFloat();
r=(int)(n+0.5f);
System.out.println("Answer="+r);
}
}

```

14. Write a program to input a floating point number and round it off to two places of decimal.

For Example:

INPUT: Enter a floating point number: 12.367

OUTPUT: Rounded off to two places of decimal:12.37

INPUT: Enter a floating point number: 14.563

OUTPUT: Rounded off to two places of decimal:14.56

Ans.

```
import java.util.*;
class Q14
{
static void main()
    {
Scanner sc=new Scanner(System.in);
float n,t,r;
System.out.println( "Enter a floating point number:");
n=sc.nextFloat();
t=(int)(n*100+0.5f);
r=t/100;
System.out.println( "Answer="+r);
    }
}
```

15. Write a program to input two integers (say a and b) and interchange their values and display the result.

For Example:

INPUT: Enter two integers: 15 36

OUTPUT:

Before interchange: a=15 and b=36

After interchange: a=36 and b=15

Please note that you can take just one variable other than a and b for interchanging.

Ans.

```
import java.util.*;
class Q15
{
```

```

static void main()
{
    Scanner sc=new Scanner(System.in);
    int a,b,c;
    System.out.println( "Enter 2 integers:");
    a=sc.nextInt();
    b=sc.nextInt();
    System.out.println("Before Interchange:a="+a+"and b="+b);
    c=a;
    a=b;
    b=c;
    System.out.println("After Interchange:a="+a+"and b="+b);
}
}

```

**16. Write a program to input the time in hours, minutes and seconds and print it in seconds.**

**For Example:**

```

INPUT:      Enter time in hours:
                1
Enter time in minutes:
                12
Enter time in seconds:
                6
OUTPUT:
Time in seconds:4326

```

**Ans.**

```

import java.util.*;
class Q16
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);

```

```

int h,m,s,sec;
System.out.println( "Enter time in hours:");
h=sc.nextInt();
System.out.println( "Enter time in minutes:");
m=sc.nextInt();
System.out.println( "Enter time in seconds:");
s=sc.nextInt();
sec=h*3600+m*60+s;
System.out.println( "Time in seconds:"+sec);
}
}

```

- 17. Write a program to input three integers and find their sum, without using the mathematical operator +.**

**Ans.**

```

import java.util.*;
class Q17
{
static void main()
{
Scanner sc=new Scanner(System.in);
int a,b,c,s;
System.out.println( "Enter 3 integers:");
a=sc.nextInt();
b=sc.nextInt();
c=sc.nextInt();
s=a+(-b)+(-c);
System.out.println( " Sum:"+s);
}
}

```

- 18. Write a program to enter the perimeter of a square and find its perimeter.**

**For Example,**

**INPUT:**

**Enter the perimeter of a square: 64**

**OUTPUT:**

**Area of the square is: 256**

**Ans.**

```
import java.util.*;
class Q18
{
static void main()
    {
Scanner sc=new Scanner(System.in);
float p,s,a;
System.out.println("Enter the perimeter of a square:");
p=sc.nextInt();
s=p/4; //finding the side
a=s*s;
System.out.println("Area of the square is:"+a);
    }
}
```

**19. Write a program to enter the length and area of a rectangle and find its perimeter.**

**For Example,**

**INPUT:**

**Enter the length of the rectangle:12**

**Enter the area of the rectangle: 72**

**OUTPUT:**

**Perimeter of the rectangle:36.0**

**Ans.**

```
import java.util.*;
class Q19
{
static void main()
    {
Scanner sc=new Scanner(System.in);
```

```

float l,b,a,p;
System.out.println("Enter the length of the rectangle:");
l=sc.nextFloat();
System.out.println("Enter the area of the rectangle:");
a=sc.nextFloat();
b=a/l; //finding the breadth
p=2*(l+b);
System.out.println("Perimeter of the rectangle:"+p);
}
}

```

20. Write a program to input the Basic Pay of an employee and find the gross pay of the employee for the following allowances and deductions.

**Dearness Allowance = 25% of 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.**

```

import java.util.*;
class Q20
{
static void main()
{
Scanner sc=new Scanner(System.in);
float bp,da,hra,pf,np,gp;
System.out.println("Enter the basic pay:");
bp=sc.nextFloat();
da=25/100f*bp;
hra=15/100f*bp;
pf=8.33f/100*bp;
np=bp+da+hra;
gp=np-pf;
}
}

```



```

        System.out.println("Gross Pay:"+gp);
    }
}

```

- 21. Write a program to input the dimensions of a rectangle and find the area of that square whose perimeter is equal to the perimeter of the rectangle.**

**Ans.**

```

import java.util.*;
class Q21
{
static void main()
    {
Scanner sc=new Scanner(System.in);
float l,b,p,s,a;
System.out.println("Enter the length and breadth of the rectangle:");
l=sc.nextFloat();
b=sc.nextFloat();
p=2*(l+b); //finding the perimeter
s=p/4; //finding the side of the square
a=s*s;
System.out.println("Area:"+a);
    }
}

```

- 22. Write a program to input an integer and if positive change it to negative and if negative change it to positive.**

**INPUT: Enter an integer: 12**

**OUTPUT: Sign changed:-12**

**INPUT: Enter an integer: -14**

**OUTPUT: Sign changed: 14**

**Ans.**

```

import java.util.*;
class Q22
{

```

```

static void main()
{
    Scanner sc=new Scanner(System.in);
    int n,s;
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    s=-n;;
    System.out.println("Sign changed:"+s);
}
}

```

- 23. If the marks obtained by a student in five different subjects are input through the keyboard, write a program to find out the aggregate marks and percentage marks obtained by the student. Assume that the maximum marks that can be obtained by a student in each subject is 100.**

**Ans.**

```

import java.util.*;
class Q23
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        float m1,m2,m3,m4,m5,agg,per;
        System.out.println("Enter the marks:");
        m1=sc.nextFloat();
        m2=sc.nextFloat();
        m3=sc.nextFloat();
        m4=sc.nextFloat();
        m5=sc.nextFloat();
        agg=m1+m2+m3+m4+m5;
        per=agg/500*100;
        System.out.println("Aggregate:"+agg);
        System.out.println("Percentage:"+per);
    }
}

```

```
}
```

24. If the total selling price of 15 items and the total profit earned on them is input through the keyboard, write a program to find the cost price of one item.

**Ans.**

```
import java.util.*;
class Q24
{
static void main()
    {
Scanner sc=new Scanner(System.in);
float tsp,tp,tcp,cp;
System.out.println("Enter the total selling price:");
tsp=sc.nextFloat();
System.out.println("Enter the total profit:");
tp=sc.nextFloat();
tcp=tsp-tp; //total cost price
cp=tcp/15; //cost price of one item
System.out.println("Cost Price of one item:"+cp);
    }
}
```

# Chapter 7

## Mathematical Library

### Methods



#### A. Tick (✓) the correct option.

1. Which among the following package is imported by default:

- a. java.lang
- b. java.util
- c. Both a and b
- d. None of these

**Ans.** a. java.lang

2. Which among the following class do not belong to the java.lang package?

- a. System
- b. String
- c. Math
- d. Scanner

**Ans.** d. Scanner

3. Which among the following function returns the absolute value of a number?

- a. Math.absoulute( )
- b. Math.neutral( )
- c. Math.abs( )
- d. Math.positive( )

**Ans.** c. Math.abs( )

4. Which among the following will yield -1 for negative number and 1 for positive number for a number stored in a variable a?

- a. Math.abs(a)\*a
- b. Math.abs(a) / a
- c. Math.ceil(a) / Math.floor(a)
- d. None of these

**Ans.** b. Math.abs(a) / a

5. Which among the following gives the next mathematical integer?

- a. Math.floor( )
- b. Math.ceil( )
- c. Math.random( )
- d. All of these

**Ans.** b. Math.ceil( )

6. What is the return type of Math.min(1,2.0f)?

- a. float
- b. double
- c. int
- d. long

**Ans.** a. float



## SECTION A

### A. Answer the following questions.

1. Name the class that is used for different mathematical functions. Give an example of a mathematical function.

**Ans.** class: Math

Functions: Math.pow(), Math.sqrt(), Math.floor(), etc.

2. Give the output of the following expressions.

i. If  $x = -9.99$ , calculate `Math.abs(x)`;

ii. If  $x = 9.0$ , calculate `Math.sqrt(x)`;

**Ans.** i. 9.99

ii. 3.0

### B. Give the output of the following functions:

1. `Math.floor(-126.349)`

**Ans.** -127.0

2. `Math.max(45.6,17.3)`

**Ans.** 45.6

3. `Math.min(-0.0,0.0)`

**Ans.** -0.0

4. `Math.pow(4,3)`

**Ans.** 64.0

5. `Math.sqrt(625)`

**Ans.** 25.0

6. `Math.cbrt(125)`

**Ans.** 5.0

7. `Math.max(11,11)`

**Ans.** 11

8. `Math.ceil(-12.56)`

**Ans.** -12.0

9. `Math.floor(15.36)`

**Ans.** 15.0

10. **Math.round(146.5)**

**Ans.** 147.0

**C. Write equivalent Java expressions for the following:**

i.  $\sqrt{a+b}$

**Ans.** Math.sqrt(a+b)

ii.  $\frac{1}{3}a^3 + \frac{1}{4}b^3$

**Ans.** 1/3.0\*Math.pow(a,3)+1/4.0\*Math.pow(b,3)

iii.  $s = ut + \frac{1}{2}ft^2$

**Ans.** s=u\*t+1/2.0\*a\*Math.pow(t,2);

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

**Ans.** d=Math.sqrt(l\*l+b\*b);

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

**Ans.** (-b+Math.sqrt(b\*b-4\*a\*c))/(2\*a)

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

**Ans.** (Math.pow(a,3)+Math.pow(b,3))/(Math.pow(a,3)-Math.pow(b,3))

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

**Ans.** Math.abs((a-b)/(a+b))

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

**Ans.** Math.cbrt((a+b)/(a\*b))

ix.  $\frac{\sqrt[3]{5x^2 + y}}{\sqrt[4]{x+1} y^3}$

**Ans.** Math.cbrt(5\*x\*x+y)/Math.pow(x+1\*y\*y\*y, 1/4.0)

x.  $\frac{x+y}{\sqrt{|x-y|}}$

**Ans.** (x+y)/Math.sqrt(Math.abs(x-y))

## SECTION B

Write programs for the following:

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

**Ans.**

```
import java.util.*;
class Q1
{
static void main()
    {
double a,p,s;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the area of the 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.

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

**Ans.**

```
import java.util.*;
class Q2
{
static void main()
    {
double l,b,d;
Scanner sc=new Scanner(System.in);
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 that outputs the results of the following evaluations based on the number entered by the user.
- i. Cube root of the number
  - ii. Absolute value of the number
  - iii. Square root of the number
  - iv. Random numbers between 0 and 1.

**Ans.**

```

import java.util.*;
class Q3
{
    static void main()
    {
        double n,c,a,s,r;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number:");
        n=sc.nextDouble();
        c=Math.cbrt(n);
        a=Math.abs(n);
        s=Math.sqrt(n);
        r=Math.random();
        System.out.println("Cube Root="+c);
        System.out.println("Absolute="+a);
        System.out.println("Square Root="+s);
        System.out.println("Random number="+r);
    }
}

```

4. Write a program to input the radius of a circle and find its area and circumference.

$$\text{Area} = \frac{22}{7} * \text{radius}^2 \quad \text{Circumference} = 2 * \frac{22}{7} * \text{radius}$$

**Ans.**

```

import java.util.*;
class Q4
{

```

```

static void main()
{
double c,a,r;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the radius of the circle:");
r=sc.nextDouble();
a=22/7.0 * Math.pow(r,2);
c=2*22/7.0*r;
System.out.println("Area="+a);
System.out.println("Circumference="+c);
}
}

```

**5. Write a program to input the area of a circle and find its circumference.**

**Ans.**

```

import java.util.*;
class Q5
{
static void main()
{
double c,a,r;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the area of the circle:");
a=sc.nextDouble();
r=Math.sqrt(a*7/22);
c=2*22/7.0*r;
System.out.println("Circumference="+c);
}
}

```

**6. Write a program to input three integers and find the sum of their cube roots.**

**Ans.**

```

import java.util.*;
class Q6

```

```

{
static void main()
    {
    double s;
    int a,b,c;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter 3 integers:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    s=Math.cbrt(a)+Math.cbrt(b)+Math.cbrt(c);
    System.out.println("Sum="+s);
    }
}

```

7. Write a program to input the principal, rate and time and find the Simple Interest.

$$\text{Simple Interest} = \frac{\text{principal} * \text{rate} * \text{time}}{100}$$

**Ans.**

```

import java.util.*;
class Q7
{
static void main()
    {
    double si,p,r,t;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter principal, rate and time:");
    p=sc.nextDouble();
    r=sc.nextDouble();
    t=sc.nextDouble();
    si=(p*r*t)/100;
    System.out.println("Simple Interest="+si);
    }
}

```

8. Write a program to input the principal, rate and time and find the compound interest.

$$\text{Compound Interest} = \text{Principal} \left(1 + \frac{\text{Rate}}{100}\right)^{\text{time}} - \text{Principal}$$

Ans.

```
import java.util.*;
class Q8
{
static void main()
    {
double ci,p,r,t;
Scanner sc=new Scanner(System.in);
System.out.println("Enter principal, rate and time:");
p=sc.nextDouble();
r=sc.nextDouble();
t=sc.nextDouble();
ci=p*Math.pow(1+r/100,t)-p;
System.out.println("Compound Interest="+ci);
    }
}
```

9. Write a program to input the three sides of a triangle and find the perimeter and area of the triangle.

If a, b and c are the 3 sides then,

Perimeter=a+b+c

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$$

where  $s = \frac{a+b+c}{2}$ .

Ans.

```
import java.util.*;
class Q9
{
static void main()
    {
double a,b,c,s,ar,p;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the 3 sides of the triangle:");
```

```

a=sc.nextDouble();
b=sc.nextDouble();
c=sc.nextDouble();
p=a+b+c;
s=p/2;
ar=Math.sqrt(s*(s-a)*(s-b)*(s-c));
System.out.println("Area="+ar);
}
}

```

10. Write a program to input a real number(floating point number) and round it off to 2 places of decimal.

For example,

**Input: 4.3678**

**Output: 4.37**

**Input: 24.3123**

**Output: 24.31**

**Ans.**

```

import java.util.*;
class Q10
{
static void main()
{
double n,r;
Scanner sc=new Scanner(System.in);
System.out.println("Enter a real number:");
n=sc.nextDouble();
r=Math.round(n*100)/100.0;
System.out.println("Rounded off to 2 places of decimal="+r);
}
}

```

11. Write a program to input three integers and print the largest and the smallest number using `Math.max()` and `Math.min()` function.

**Ans.**

```
import java.util.*;
class Q11
{
static void main()
    {
    int a,b,c,l,s;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter 3 integers:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    l=Math.max(a,Math.max(b,c));
    s=Math.min(a,Math.min(b,c));
    System.out.println("largest="+l);
    System.out.println("smallest="+s);
    }
}
```

12. A number  $n$  is said to be a Sunny number if,  $\sqrt{n+1}$  is equal to an integer. For example, 8 is sunny as  $\sqrt{8+1} = 3$  which is an integer. Write a program to input a number and using ternary operator check whether it is a sunny number or not.

**Ans.**

```
import java.util.*;
class Q12
{
static void main()
    {
    int n;
    double s;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter 3 integers:");
```

```

        n=sc.nextInt();
        s=Math.sqrt(n+1);
        System.out.println((s==(int)s)? "Sunny Number": "Not a Sunny Number");
    }
}

```

**13. Write a program to input a number and display 1, if it is positive and – 1 , if it is negative.**

**Ans.**

```

import java.util.*;
class Q13
{
static void main()
    {
        int n,s;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter an integer:");
        n=sc.nextInt();
        s=Math.abs(n)/n;
        System.out.println(s);
    }
}

```

**14. Write a program to input positive real number and display its Math.ceil( ) value. Note that the function Math.ceil( ) should not be used.**

**Ans.**

```

import java.util.*;
class Q14
{
static void main()
    {
        double n,c;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a real number:");
        n=sc.nextDouble();
        c=(int)(n+0.99999999);
    }
}

```

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

**15. Write a program to input three numbers and find the sum of their square roots.**

**Ans.**

```
import java.util.*;
class Q15
{
    static void main()
    {
        double a,b,c,s;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter 3 numbers:");
        a=sc.nextDouble();
        b=sc.nextDouble();
        c=sc.nextDouble();
        s=Math.sqrt(a)+Math.sqrt(b)+Math.sqrt(c);
        System.out.println("Sum="+s);
    }
}
```

**16. Write a program to input three numbers and find the average of their cube roots.**

**Ans.**

```
import java.util.*;
class Q16
{
    static void main()
    {
        double a,b,c,av;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter 3 numbers:");
        a=sc.nextDouble();
        b=sc.nextDouble();
```



```

        c=sc.nextDouble();
        av=(Math.cbrt(a)+Math.cbrt(b)+Math.cbrt(c))/3;
        System.out.println("Average="+av);
    }
}

```

- 17. Write a program to input positive real number and display its Math.floor( ) value. Note that the function Math.floor( ) should not be used.**

**Ans.**

```

import java.util.*;
class Q17
{
static void main()
    {
        double n,f;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number:");
        n=sc.nextDouble();
        f=(int)n;
        System.out.println(f);
    }
}

```

- 18. Write a program to generate 2 random integers between 10 to 99 (both inclusive) and find their sum, product and difference.**

**Ans.**

```

class Q18
{
static void main()
    {
        int r1,r2,s,d,p;
        r1=10+(int)(Math.random()*(99-10+1));
        r2=10+(int)(Math.random()*(99-10+1));
        s=r1+r2;
        d=r1-r2;
    }
}

```

```

    p=r1*r2;
    System.out.println("Sum="+s);
    System.out.println("Difference="+d);
    System.out.println("Product="+p);
    }
}

```

- 19. Write a program to input the length and breadth of a rectangle and find the perimeter of that square whose area is same as the area of the rectangle.**

**Ans.**

```

import java.util.*;
class Q19
{
static void main()
    {
    double l,b,a,s,p;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the length and breadth of the rectangle:");
    l=sc.nextDouble();
    b=sc.nextDouble();
    a=l*b;
    s=Math.sqrt(a);
    p=4*s;
    System.out.println("Perimeter="+p);
    }
}

```

- 20. Write a program to input three integers and find the sum of the cube roots of their last digits.**

**Ans.**

```

import java.util.*;
class Q20
{
static void main()
    {
    double s;

```

```
int a,b,c,l1,l2,l3;  
Scanner sc=new Scanner(System.in);  
System.out.println("Enter 3 integers:");  
a=sc.nextInt();  
b=sc.nextInt();  
c=sc.nextInt();  
l1=a%10;  
l2=b%10;  
l3=c%10;  
s=Math.cbrt(l1)+Math.cbrt(l2)+Math.cbrt(l3);  
System.out.println("Sum="+s);  
}  
}
```

# Chapter 8

## Conditional Constructs in Java



Tick (✓) the correct option.

1. Which among the following constants are used in switch-case statements?
- a. char
  - b. int
  - c. float
  - d. Both a and b

**Ans.** d. Both a and b

2. If a relational expression is satisfied it results in \_\_\_\_\_.
- a. true
  - b. false
  - c. Both a and b
  - d. None of these

**Ans.** a. true

3. Which among the following relational operator is used to check for the equality between two quantities ?
- a. =
  - b. ==
  - c. equals
  - d. None of these

**Ans.** b. ==

4. Which among the following logical operator is used to negate a condition?
- a. &&
  - b. ||
  - c. !
  - d. All of these

**Ans.** c. !

5. An if statement inside another if statement is called \_\_\_\_\_.
- a. Nested if
  - b. Internal if
  - c. Concentric if
  - d. None of these

**Ans.** a. Nested if

6. What is the problem called in which an optional else clause in an if-then(-else) statement results in nested conditionals being ambiguous?
- a. Dangling if
  - b. Dangling else
  - c. Both a and b
  - d. None of these

**Ans.** b. Dangling else

7. Which of these selection statements test only for equality?
- a. if
  - b. switch
  - c. if and switch
  - d. none of the mentioned

**Ans.** b. switch

8. State whether the following statements about switch statement are correct.
- i. Switch statement often provides a better alternative than a large series of if-else-if statements.
  - ii. The break statement is used inside the switch to terminate a statement sequence.
- a. True, False
  - b. False, True
  - c. True, True
  - d. False, False

**Ans.** c. True, True

9. The conditional statement, can only test for equality, whereas can evaluate any type of Boolean expression.
- a. if, switch
  - b. switch, if
  - c. while, if
  - d. if, while

**Ans.** b. switch, if

10. What will be the output of the following code snippet?

```
int a=15;
int b=25;
if ((a<b) || (a=5)>15)
system.out.println(a);
else
system.out.println(b);
```

- a. Error
- b. 15
- c. 25
- d. No output

**Ans.** b. 15

## SECTION A

### I. Answer the following questions.

1. **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.

Conditional statements in Java are : if, if- else, if –else if –else and switch case.

**2. When are braces optional or necessary in conditional statements?**

**Ans.** If only one statement is to be executed depending upon a condition, giving braces is optional.

**3. What are relational operators?**

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

**4. State the difference between = and ==.**

**Ans.** The = is an assignment operator and is used to assign a value to a variable. The == is a relational operator and is used to check for equality.

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

**Ans.** Logical Operators are used to join more than one *relational expression*. The following statements explain the usage of logical operators:

- If I have a pen AND a paper it is possible for me to write. This is an example of AND join, where both the conditions need to be satisfied for the resultant to be true. Thus only when I have both, pen and paper it will be possible for me to write. Thus AND operator is used when all conditions need to be satisfied.
- If I have a pen OR a pencil it is possible for me to write. This is an example of OR join, where either of the conditions need to be satisfied for the resultant to be true. Thus if I have a pen I can write or if I have a pencil still I can write. Thus OR operator is used when either of the conditions need to be satisfied.
- If it is NOT that, I do not have a pen, it is possible for me to write. Here see how the NOT operator is used to negate a condition.

**6. What is the function of nested if?**

**Ans.** A nested if is an if statement that is the target of another if statement. Nested if statements means an if statement inside another if statement. This is generally used when a condition needs to be satisfied inside another condition.

**7. What is a conditional operator? Explain with the help of an example.**

**Ans.** This operator is used as an alternative to if statement. The ? and : are used as conditional or ternary operator. This operator can be used to replace if else statements of the general form:

if (condition)

statement1;

else

statement2;

The above form of if can be alternatively written using ? and : as follows:

condition? statement1 : statement2;

It works in the same way as the above given form of if. That is test-condition is evaluated and if it is true, statement1 gets executed (i.e. its value becomes the value of entire expression) otherwise statement2 gets executed (i.e. its value now becomes the value of the entire expression). For

instance, the following if statement

```
int c;  
if (a>b)  
    c=a;  
else  
    c=b;
```

can be alternatively written as

```
int c;  
c = ((a>b) ? a : b);
```

### 8. Compare if and ? :.

**Ans.**

If-else	Ternary Operator
It is used when complex logic handling is necessary.	It is used to offer more concise, clean and compact code
It can have multiple statements, multiple assignments and expressions (more than one statements) in its body.	It produces an expression, and hence a single value can, be assigned or incorporated into a larger expression.
Nested if statement is simple and fairly easy to understand.	Ternary operator in its nested form, it becomes complex and difficult to understand.

### 9. What is the function of switch statement?

**Ans.** A switch statement is a type of selection control mechanism used to allow the value of a variable or expression to change the control flow of program execution via a multiway branch.

### 10. State the rules, which is to be followed while using switch statement.

**Ans.** The following rules are to be kept in mind while using switch-case statements:

- A **case** statement cannot exist by itself, outside of a **switch** block.
- The **break** statement which appears as the last statement in a case block is one of Java's jump statements. When a **break** statement is encountered in the **case** block, program execution jumps to the line of code following the **switch** block i.e. outside the body of **switch** statement. If **break** statement is missing in a case block upon completion of its execution, the control goes to the following case block, this process continues until break statement is encountered or **switch** block ends.
- Braces should be used to denote the start and end of the **switch** block.
- Only character constants and Integer constants are used in the **case** constants but allows only for equality comparisons.
- All **case** labels in the same switch block may have unique values.

11. Explain, with the help of an example, the purpose of default in a switch statement.

Ans. The default block gets executed when none of the case matches with the switch expression.

12. Differentiate between if and switch statements.

Ans.

If-else	Switch case
It checks for both relational and logical operations	It checks only for equality.
I may be used with almost any data type.	It is used only with integers and characters.
Comparison may be a variable or an expression or constant with another variable, expression or constant.	Comparison is always with a variable or expression with an integer or character constant.

13 Explain with an example the if-else-if construct.

Ans. The following example shows the formation of grade depending upon the marks obtained by students in a school.

Marks	Grade
90 to 100	A
70 to 89	B
50 to 69	C
30 to 49	D
below 30	E

The corresponding code for the grading system would be:

```
if (marks >= 90)
    System.out.println("Grade A");
else if (marks >= 70)
    System.out.println("Grade B");
else if (marks >= 50)
    System.out.println("Grade C");
else if (marks >= 30)
    System.out.println("Grade D");
else
    System.out.println("Grade E");
```

In this code if someone gets **95** the condition **marks >= 90** gets satisfied and **Grade A** gets printed. Again if **marks** is **85** the **marks >= 70**, **marks >= 50** and **marks >= 30** all gets satisfied but only **marks >= 70** will get executed as this the first satisfying statement in the **else-if** ladder and therefore **Grade B** gets executed. Similarly if marks is **55**, **Grade C** gets printed. Now if **marks** is **15**, none of the above conditions gets satisfied and therefore the statement in the else part gets executed, which prints **Grade E**.



**B. Do as directed in the following questions:**

1. Using the given values of x, y and z evaluate the following (Answer in true / false).

`(x>=y) || (!(z==y) && (z<x))`

i. `x=10, y=5, z=11`

**Ans.** true

ii. `x=10, y=10, z=10`

**Ans.** true

iii. `x=9, y=10, z=2`

**Ans.** true

2. Using conditional operator write equivalent expressions for the following:

i. `F=0.5` if `x=30` otherwise `0`.

**Ans.** `F=(x==30)?0.5:0;`

ii. Print “Even” if x is an even number otherwise print “Odd”.

**Ans.** `System.out.println((x%2==0)? “Even”: “Odd”;`

iii. Rewrite the following using Ternary operator

`if (income <= 10000 )`

`tax = 0;`

`else`

`tax = 12;`

**Ans.** `tax=(income<=10000)?0:12;`

3. Find the output of the following program segment, when:

i. `val = 500`

ii. `val = 1600`

`int val, sum, n 550;`

`sum = n + val > 750? 400 : 200;`

`System.out.pritln(sum);`

**Ans.** i. 200

ii. 400

4. Write the conditional statement for the following decision making situation: Provided the value of x is greater than 9, increase the value of y by 15, otherwise decrease the value of y by 36.

**Ans.**  $y=(x>9)?y+15:y-36;$

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

```
char grade=(mark>=90)? "A" : (mark>=80)? "B" : "C";
```

**Ans.** char grade;

```
If (mark>=90)
```

```
    grade='A';
```

```
else if(mark>=80)
```

```
    grade='B';
```

```
else
```

```
    grade='C';
```

6. Write equivalent Java statements for the following situations:

i. X is greater than equal to Y.

**Ans.**  $X \geq Y$

ii. X is not equal to Y.

**Ans.**  $X \neq Y$

iii. X is equal to Y.

**Ans.**  $X == Y$

iv. X is less than Y and X is less than Z.

**Ans.**  $X < Y \ \&\& \ X < Z$

v. Either X is greater than Y or X is less than Z.

**Ans.**  $X > Y \ || \ X < Z$

vi. If X has a value greater than 20 then increment the value of J by 1 otherwise decrement the value by 2.

**Ans.**  $J=(X>20)?J+1:J-2;$

vii. It is not that X is greater than Y.

**Ans.**  $!(X>Y)$

7. The following is a segment of a program:

```
x=1; y=1;
```

```
if(n>0)
```

```
{
```

```
x=x+1;
y=y-1;
}
```

What will be the value of x and y, if n assumes a value (i) 1 (ii) 0?

- Ans.** i. x=2 and y=0  
ii. X=1 and y=1

8. What is the output of the following program:

```
class First
{
static void check()
    {
float num=110;
System.out.print((num!=-num)? "OK": "NOT");
    }
}
```

**Ans.** OK

9. What is the output of the following program:

```
class Second
{
static void check()
    {
int r, a=50, b=10;
r=(a>45) ?a : b;
System.out.print(r);
    }
}
```

**Ans.** 50

10. Rewrite the following set of if-else statement in terms of switch case statement for the following code fragment:

```
int code;
code=sc.nextInt();
if (code==1)
    System.out.print("Accountant");
else
    if (code==5 || code==6)
```

```
System.out.print("Grade IV");
else
if (code==3)
System.out.print("Financial Advisor");
```

**Ans.**

```
int code;
code=sc.nextInt();
switch(code)
{
    case 1:
        System.out.print("Accountant");
    case 5:
    case 6:
        System.out.print("Grade IV");
    case 3:
        System.out.print("Financial Advisor");
}
```

11. Find the syntax error(s), if any, in the following program and rewrite it after correction:

**Class First**

```
{
public static void main(String args[]){
int R; W=90;
R=W-50;
Switch(W)
{
    20:system.out.print(Lower Range");
    30:System.Out.print("Middle Range")
    40:System.out.Print("Higher Range);
}
}
```

**Ans.**

```
class First
{
public static void main(String args[]){
int R,W=90;
R=W-50;
```

```

switch(R)
{
case 20:
    System.out.print("Lower Range");
    break;
case 30:
    System.out.print("Middle Range")
    break;
case 40:
    System.out.print("Higher Range");
}
}

```

12. Give the output of the following switch statement:

```

switch(num)
{
case 10:
case 100:
    System.out.println("Ten and hundred");
    break;
case 1000:
case 400:
    System.out.println("Thousand and 4 hundred");
    break;
default:
    System.out.println("Invalid Input");
}

```

i. num=100

ii. num=1400

iii. num=1000

iv. num=400

**Ans.**

i. Ten and hundred

ii. Invalid Input

iii. Thousand and 4 hundred

iv. Thousand and 4 hundred

13. Given the output of the following code fragment:

```
when i. opn= 'b'      iii. opn= 'x'      iii. opn= 'a'
switch(opn)
{
case 'a':
    System.out.println("Platform Independent");
    break;
case 'b':
    System.out.println("Object Oriented");
case 'c':
    System.out.println("Robust and Secure");
    break;
default:
    System.out.println("Wrong Input");
}
```

**Ans.**

- i. Object Oriented  
Robust and Secure
- ii. Wrong Input
- iii. Platform Independent

## SECTION B

Write programs for the following:

1. Write a program to input an integer and check whether it is greater than 100, or less than 100 or equal to 100.

**Ans.**

```
import java.util.*;
class Q1
{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if(n>100)
```

```

        System.out.println("Greater than 100");
    else if(n<100)
        System.out.println("Less than 100");
    else
        System.out.println("Equal to 100");
    }
}

```

2. Write a program to input the Selling Price and Cost Price of a commodity and find the Profit or Loss made upon selling the product.

For example,

**INPUT:**

Enter the Selling Price of the commodity: 456

Enter the Cost Price of the commodity: 400

**OUTPUT:**

Profit: 56

**INPUT:**

Enter the Selling Price of the commodity: 300

Enter the Cost Price of the commodity: 310

**OUTPUT:**

Loss: 10

**Ans.**

```

import java.util.*;
class Q2
{
static void main()
    {
    int sp,cp,ans;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the Selling Price of the commodity:");
    sp=sc.nextInt();
    System.out.println("Enter the Cost Price of the commodity:");
    cp=sc.nextInt();
    if(sp>cp)
    {
    ans=sp-cp;

```

```

        System.out.println("Profit:"+ans);
    }
    else if(sp<cp)
    {
        ans=cp-sp;
        System.out.println("Loss:"+ans);
    }
    else
        System.out.println("No profit or loss");
    }
}

```

3. Write a program to check whether a int type number taken as input is a positive, negative or a zero.

**Ans.**

```

import java.util.*;
class Q3
{
    static void main()
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number:");
        n=sc.nextInt();
        if(n>0)
            System.out.println("Positive");
        else if(n<0)
            System.out.println("Negative");
        else
            System.out.println("Zero");
    }
}

```

4. Write a program to check whether a short type number taken as input is an even number or an odd number.

**Ans.**

```

import java.util.*;
class Q4

```



```

{
static void main()
    {
    short n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number:");
    n=sc.nextShort();
    if(n%2==0)
    System.out.println("Even");
    else
    System.out.println("Odd");
    }
}

```

5. Write a program to accept an int type value as a parameter and print the square of it, if it is more than 120, otherwise print its cube.

**Ans.**

```

class Q5
{
static void main(int n)
    {
    if(n>120)
    System.out.println("Square="+(n*n));
    else
    System.out.println("Cube="+(n*n*n));
    }
}

```

6. Write a program to input 5 real numbers(floating point) and find their average. If the average is greater than 50 print good, otherwise print bad.

**Ans.**

```

import java.util.*;
class Q6
{
static void main()
    {
    float a,b,c,d,e,av;
    Scanner sc=new Scanner(System.in);

```

```

System.out.println("Enter 5 real numbers:");
a=sc.nextFloat();
b=sc.nextFloat();
c=sc.nextFloat();
d=sc.nextFloat();
e=sc.nextFloat();
av=(a+b+c+d+e)/5;
System.out.println("Average="+av);
if(av>50)
System.out.println("Good");
else
System.out.println("Bad");
}
}

```

7. Write a program to check whether an int type number (taken as input) is a multiple of 5 or not.

**Ans.**

```

import java.util.*;
class Q7
{
static void main()
{
int n;
Scanner sc=new Scanner(System.in);
System.out.println("Enter an integer:");
n=sc.nextInt();
if(n%5==0)
System.out.println("Multiple of 5");
else
System.out.println("Not a multiple of 5");
}
}

```

8. Write a program to check whether an int type number (taken as input) is a 2-digit number or not.

**Ans.**

```

import java.util.*;
class Q8

```

```

{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if(n>=10 && n<=99 || n>=-99 && n<=-10)
    System.out.println("2-digit number");
    else
    System.out.println("Not a 2-digit number");
    }
}

```

9. Write a program to check whether an int type number (taken as input) is a 3-digit number divisible by three or not.

**Ans.**

```

import java.util.*;
class Q9
{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if((n>=100 && n<=999 || n>=-999 && n<=-100) && n%3==0)
    System.out.println("3-digit number divisible by 3");
    else
    System.out.println("Not a 3-digit number or divisible by 3");
    }
}

```

10. Write a program to input an integer and check whether it is a 2-digit or a 3-digit number or not. In case it is not a 2-digit or a 3-digit number a relevant message ("Not 2 or 3 digit number") should be displayed.

**Ans.** import java.util.\*;

```

class Q10
{

```

```

static void main()
{
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if(n>=10 && n<=99 || n>=-99 && n<=-10)
    System.out.println("2-digit number");
    else if(n>=100 && n<=999 || n>=-999 && n<=-100)
    System.out.println("3-digit number");
    else
    System.out.println("Not a 2 or 3-digit number");
}
}

```

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

**Ans.**

```

import java.util.*;
class Q11
{
    static void main()
    {
        int a,b;
        Scanner sc=new Scanner(System.in);
        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");
    }
}

```

- 12. Write a program to input three integers and check whether all of them are unique (different) from each other or not.**

**Ans.**

```

import java.util.*;
class Q12
{

```

```

static void main()
{
    int a,b,c;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter 3 integers:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    if(a!=b && b!=c && a!=c)
        System.out.println("All are unique");
    else
        System.out.println("All are not unique");
}
}

```

- 13. 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 Q13
{
    static void main(int a, int b)
    {
        float c;
        if (a==0 || b==0)
            System.out.println("Invalid entry");
        else
        {
            if(a>b)
                c=(float)a/b;
            else
                c=(float)b/a;
            System.out.println("Quotient:"+c);
        }
    }
}
}

```

14. 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.

```
class Q14
{
static void main(int a,int b,int c)
{
if(a<b+c && b<c+a & c<a+b)
{
if(a*a<b*b+c*c && b*b<c*c+a*a && c*c<a*a+b*b)
System.out.println("Acute angled triangle");
else if(a*a>b*b+c*c || b*b>c*c+a*a || c*c>a*a+b*b)
System.out.println("Obtuse angled triangle");
else
System.out.println("Right angled triangle");
}
else
System.out.println("Does not form a triangle");
}
}
```

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

<u>Marks</u>	<u>Grade</u>
Above 90	A
70 to 89	B
50 to 69	C
below 50	D

**Ans.**

```
import java.util.*;
class Q15
{
static void main()
    {
    int m;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the marks:");
    m=sc.nextInt();
    if (m>=90)
    System.out.println("Grade=A");
    else if (m>=70)
    System.out.println("Grade=B");
    else if (m>=50)
    System.out.println("Grade=C");
    else
    System.out.println("Grade=D");
    }
}
```

16. 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 Q16
{
static void main()
    {
```

```

int tc;
float dis,r,ap;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the total cost:");
tc=sc.nextInt();
if (tc<=2000)
r=5;
else if (tc<=5000)
r=25;
else if (tc<=10000)
r=35;
else
r=50;
dis=r/100*tc;
ap=tc-dis;
System.out.println("Amount Payable="+ap);
}
}

```

17. An electricity company charges their consumers according to the units consumed per month according to the given traffic:

Units Consumed	Charges
Up to 100 units	₹ 2 per unit
More than 100 units and up to 200 units	₹ 1.80 per unit
More than 200 units	₹ 1.50 per unit

In addition to the above, every consumer has to pay ₹ 200 as Service Charge per month. Write a program to input the amount of units consumed and calculate the total charges payable (Bill) by the consumer.

**Ans.**

```

import java.util.*;
class Q17
{
static void main()
{
float u,c;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the units consumed:");
u=sc.nextInt();

```



```

    if (u<=100)
    c=2*u;
    else if (u<=200)
    c=100*2+(u-100)*1.8f;
    else
    c=100*2+100*1.8f+(u-200)*1.50f;
    c=200+c; //Add the service charge
    System.out.println("Amount Payable="+c);
    }
}

```

**18. A library charges a fine for books returned late. Following are the fines:**

<b>First five days</b>	<b>:</b>	<b>40 paise per day.</b>
<b>Six to Ten days</b>	<b>:</b>	<b>65 paise per day.</b>
<b>Above ten days</b>	<b>:</b>	<b>80 paise per day.</b>

**Design a program to calculate the fine assuming that a book is returned N days late.**

**Ans.**

```

import java.util.*;
class Q18
{
static void main()
    {
    int N;
    float f;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the no. of days the book is returned late:");
    N=sc.nextInt();
    if (N<=5)
    f=0.40f*N;
    else if (N<=10)
    f=0.40f*5+(N-5)*0.65f;
    else
    f=0.40f*5+0.65f*5+(N-10)*0.80f;
    System.out.println("Fine="+f);
    }
}

```

19. Write a program to input an integer and check whether it is a 5-digit number or not. If it is extract the central digit and print it.

**Example**

**INPUT: Enter an integer: 76549**

**OUTPUT: Central digit: 5**

**INPUT: Enter an integer: 12976**

**OUTPUT: Central digit: 9**

**INPUT: Enter an integer: 126**

**OUTPUT: Not a 5 digit number**

**Ans.**

```
import java.util.*;
class Q19
{
static void main()
    {
    int n,m;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if (n>=10000 && n<=99999)
    {
    m=(n/100)%10;
    System.out.println("Central digit:"+m);
    }
    else
    System.out.println("Not a 5 digit number");
    }
}
```

20. Write a program to input an integer and check whether it is a 4-digit number or not. If it is find the sum of first and last digit.

**Example**

**INPUT: Enter an integer: 4765**

**OUTPUT: Sum of the first and last digit: 9**

**INPUT: Enter an integer: 7839**

**OUTPUT: Sum of the first and last digit: 16**

**INPUT: Enter an integer: 126**

**OUTPUT: Not a 4 digit number**

**Ans.**

```
import java.util.*;
class Q20
{
static void main()
    {
    int n,f,l,s;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if (n>=1000 && n<=9999)
    {
    f=n/1000;
    l=n%10;
    s=f+l;
    System.out.println("Sum of the first and last digit:"+s);
    }
    else
    System.out.println("Not a 4digit number");
    }
}
```

21. Write a program to input an integer and check whether it is a 3-digit number or not. If it is check whether the sum of first and last digit is Even or an Odd number.

**Example**

**INPUT: Enter an integer: 476**

**OUTPUT: Sum of the first and last digit is an Even number**

**INPUT: Enter an integer: 784**

**OUTPUT: Sum of the first and last digit is an Odd number**

**INPUT: Enter an integer: 12**

**OUTPUT: Not a 3 digit number**

**Ans.**

```
import java.util.*;
class Q21
```

```

{
static void main()
    {
    int n,f,l,s;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if (n>=100 && n<=999)
    {
    f=n/100;
    l=n%10;
    s=f+l;
    if (s%2==0)
    System.out.println("Sum of the first and last digits is an Even number");
    else
    System.out.println("Sum of the first and last digits is an Odd number");
    }
    else
    System.out.println("Not a 3 digit number");
    }
}

```

22. 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-digit number" otherwise, output the message "Not a special 2-digit number".**

**Ans.**

```

import java.util.*;
class Q22
{
static void main()
    {

```

```

int n,f,l,s,p;
Scanner sc=new Scanner(System.in);
System.out.println("Enter an integer:");
n=sc.nextInt();
if (n>=10 && n<=99)
{
f=n/10;
l=n%10;
s=f+l;
p=f*l;
if (s+p==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");
}
}

```

- 23. Write a program to input an integer and using ternary operator check whether it is an Even or an Odd number.**

**Ans.**

```

import java.util.*;
class Q23
{
static void main()
{
int n;
Scanner sc=new Scanner(System.in);
System.out.println("Enter an integer:");
n=sc.nextInt();
System.out.println((n%2==0)?"Even":"Odd");
}
}

```

**24. Write a program to input an integer and using ternary operator check whether it is a multiple of 5 or not.**

**Ans.**

```
import java.util.*;
class Q24
{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    System.out.println((n%5==0)?"Multiple of 5":"Not a multiple of 5");
    }
}
```

**25. Write a program to input an integer and using ternary operator check whether it is a positive number , negative number or a zero or not.**

**Ans.**

```
import java.util.*;
class Q25
{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    System.out.println((n>0)?"Positive":(n<0)?"Negative":"Zero");
    }
}
```

**26. Using a switch statement, write a menu driven program to convert a given temperature from Fahrenheit to Celsius and vice versa. For an incorrect choice, an appropriate error message should be displayed.**

**(HINT :  $C = \frac{5}{9} \times (F - 32)$  and  $F = 1.8 \times C + 32$ )**

**Ans.**

```
import java.util.*;
class Q26
```

```

{
static void main()
    {
    int ch;
    float f,c;
    Scanner sc=new Scanner(System.in);
    System.out.println("MENU");
    System.out.println("1.Fahrenheit to Celcius");
    System.out.println("2.Celcius to Farenheit");
    System.out.println("Enter your choice:");
    ch=sc.nextInt();
    switch(ch)
    {
    case 1:
    System.out.println("Enter the temperature in farenheit:");
    f=sc.nextFloat();
    c=5/9f*(f-32);
    System.out.println("Celcius="+c);
    break;
    case 2:
    System.out.println("Enter the temperature in celcius:");
    c=sc.nextFloat();
    f=1.8f*c+32;
    System.out.println("Farenheit="+f);
    break;
    default:
    System.out.println("Invalid choice");
    }
    }
}

```

27. Using a switch statement, write a menu driven program to find the area and perimeter of a square or rectangle. For finding the area and perimeter of a square input the side length and for finding the area and perimeter of the rectangle input the length and breadth.

**Ans.**

```

import java.util.*;
class Q27
{

```

```

static void main()
{
    int ch,s,a,p,l,b;
    Scanner sc=new Scanner(System.in);
    System.out.println("MENU");
    System.out.println("1.Area/Perimeter of a square");
    System.out.println("2.Area/Perimeter of a rectangle");
    System.out.println("Enter your choice:");
    ch=sc.nextInt();
    switch(ch)
    {
        case 1:
            System.out.println("Enter the side of a square:");
            s=sc.nextInt();
            a=s*s;
            p=4*s;
            System.out.println("Area="+a);
            System.out.println("Perimeter="+p);
            break;
        case 2:
            System.out.println("Enter the length and breadth of a rectangle:");
            l=sc.nextInt();
            b=sc.nextInt();
            a=l*b;
            p=2*(l+b);
            System.out.println("Area="+a);
            System.out.println("Perimeter="+p);
            break;
        default:
            System.out.println("Invalid choice");
    }
}
}

```

**28. Write a menu driven program to find:**

- a. Area of a rectangle, where length and breadth is taken as input.(Area=length \*breadth)
- b. Area of a circle, where radius is taken as input.(Area=  $22/7$ \*radius<sup>2</sup>)

**Ans.**

```

import java.util.*;
class Q28

```



```

{
static void main()
    {
    int ch,a,l,b;
    float ar,r;
    Scanner sc=new Scanner(System.in);
    System.out.println("MENU");
    System.out.println("1.Area of a rectangle");
    System.out.println("2.Area of a circle");
    System.out.println("Enter your choice:");
    ch=sc.nextInt();
    switch(ch)
    {
    case 1:
    System.out.println("Enter the length and breadth of a rectangle:");
    l=sc.nextInt();
    b=sc.nextInt();
    a=l*b;
    System.out.println("Area="+a);
    break;
    case 2:
    System.out.println("Enter the radius of a circle:");
    r=sc.nextFloat();
    ar=22/7f*r*r;
    System.out.println("Area="+ar);
    break;
    default:
    System.out.println("Invalid choice");
    }
    }
}

```

29. Write a menu driven program to input a day number between 1 to 7 and print the corresponding weekday. That is for 1 display Sunday, for 2 display Monday for 3 display Tuesday, etc.

**Ans.**

```

import java.util.*;
class Q29
{

```

```

static void main()
{
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number in the range 1-7:");
    n=sc.nextInt();
    switch(n)
    {
        case 1:
            System.out.println("Sunday");
            break;
        case 2:
            System.out.println("Monday");
            break;
        case 3:
            System.out.println("Tuesday");
            break;
        case 4:
            System.out.println("Wednesday");
            break;
        case 5:
            System.out.println("Thursday");
            break;
        case 6:
            System.out.println("Friday");
            break;
        case 7:
            System.out.println("Saturday");
            break;
        default:
            System.out.println("Invalid input");
    }
}

```

- 30. Write a menu driven program to input a month number between 1 to 12 and print the corresponding month name. That is for 1 display January, for 2 display February for 3 display March, etc.**

**Ans.**

```

import java.util.*;
class Q30

```

```

{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number in the range 1-12:");
    n=sc.nextInt();
    switch(n)
    {
    case 1:
    System.out.println("January");
    break;
    case 2:
    System.out.println("February");
    break;
    case 3:
    System.out.println("March");
    break;
    case 4:
    System.out.println("April");
    break;
    case 5:
    System.out.println("May");
    break;
    case 6:
    System.out.println("June");
    break;
    case 7:
    System.out.println("July");
    break;
    case 8:
    System.out.println("August");
    break;
    case 9:
    System.out.println("September");
    break;
    case 10:
    System.out.println("October");
    break;

```

```

    case 11:
    System.out.println("November");
    break;
    case 12:
    System.out.println("December");
    break;
    default:
    System.out.println("Invalid input");
    }
}
}

```

- 31. Write a program to input an integer and check whether it is divisible by 3 or not using switch case only (no if-else statement).**

**Ans.**

```

import java.util.*;
class Q31
{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number in the range 1-12:");
    n=sc.nextInt();
    switch(n%3)
    {
    case 0:
    System.out.println("Divisible by 3");
    break;
    default:
    System.out.println("Not divisible by 3");
    }
}
}

```

- 32. The 1st day of 2010 was Friday, write a program to input any day number within the month of January and print which day was it. Also check whether the day number entered is a valid date or not.**

**For example,**

**Input: Enter day number: 20**

**Output: Wednesday**

**Input: Enter day number:30**

**Output: Saturday**

**Input: Enter day number:34**

**Output: Invalid day number**

**Ans.**

```
import java.util.*;
class Q32
{
static void main()
    {
    int d;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a day number of the month of January:");
    d=sc.nextInt();
    if(d>=1 && d<=7)
    {
    switch(d%7)
    {
    case 1:
    System.out.println("Friday");
    break;
    case 2:
    System.out.println("Saturday");
    break;
    case 3:
    System.out.println("Sunday");
    break;
    case 4:
    System.out.println("Monday");
    break;
    case 5:
    System.out.println("Tuesday");
    break;
    case 6:
    System.out.println("Wednesday");
```

```

    break;
    case 0:
    System.out.println("Thursday");
    break;
    default:
    System.out.println("Invalid input");
    }
    }
    else
    System.out.println("Invalid day number");
    }
}

```

- 33. Write a program to input an integer and check whether it is a magic number or not. A Magic number is such a number whose continuous sum of the digits until it forms a single digit is 1. For example [55]=5+5=[10]=1+0=[1], therefore 55 is a Magic Number. (Remember no loops is to be used).**

**Ans.** `import java.util.*;`

```

class Q33
{
static void main()
    {
    int n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter an integer:");
    n=sc.nextInt();
    if(n%9==1)
    System.out.println("Magic number");
    else
    System.out.println("Not a magic number");
    }
}

```

- 34. Write a menu-driven program to check whether a number is i) Even or odd, ii) Multiple of 17 or not, iii) Two digit number or not.**

**Ans.**

```

import java.util.*;
class Q34

```

```

{
static void main()
    {
    int ch,n;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number:");
    n=sc.nextInt();
    System.out.println("MENU");
    System.out.println("1.Even or odd");
    System.out.println("2.Multiple of 17 or not");
    System.out.println("3.2 digit number or not");
    System.out.println("Enter your choice:");
    ch=sc.nextInt();
    switch(ch)
    {
    case 1:
    if(n%2==0)
    System.out.println("Even");
    else
    System.out.println("Odd");
    break;
    case 2:
    if(n%17==0)
    System.out.println("Multiple of 17");
    else
    System.out.println("Not a multiple of 17");
    break;
    case 3:
    if(n>=10 && n<=99)
    System.out.println("2 digit number");
    else
    System.out.println("Not a 2-digit number");
    break;
    default:
    System.out.println("Invalid choice");
    }
    }
}

```

**35. Write a program to input 5 real numbers and print only those numbers which are less than their average.**

**Ans.**

```
import java.util.*;
class Q35
{
static void main()
    {
float a,b,c,d,e,avg;
Scanner sc=new Scanner(System.in);
System.out.println("Enter 5 real numbers:");
a=sc.nextFloat();
b=sc.nextFloat();
c=sc.nextFloat();
d=sc.nextFloat();
e=sc.nextFloat();
avg=(a+b+c+d+e)/5;
if (a<avg)
System.out.println(a+"is less than its average");
if (b<avg)
System.out.println(b+"is less than its average");
if (c<avg)
System.out.println(c+"is less than its average");
if (d<avg)
System.out.println(d+"is less than its average");
if (e<avg)
System.out.println(e+"is less than its average");
    }
}
```

**36. Write a program to input three integers and check whether it forms a Pythagorean triplet or not. A set of three integers is said to be a Pythagoren triplet if the sum of the squares of any two integers is equal to square of the third integer. Example, (3, 4, 5), (5, 12, 13), and (7, 24, 25).**

**Ans.**

```
import java.util.*;
class Q36
{
```



```

static void main()
{
    int a,b,c;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter 3 integers:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    if (a*a==b*b+c*c || b*b==a*a+c*c || c*c==a*a+b*b)
    System.out.println("Pythagorean Triplet");
    else
    System.out.println("Not a Pythagorean triplet");
}
}

```

37. To foster a sense of water conservation, the water department has an annual water conservation tax policy. The tax are based on the water consumption of the consumer. The tax rates are as follows:

Water Consumed (in Gallons)	Tax Rate in ₹ /100 gallons
Upto 45	No Tax
More than 45 but 75 or less	475.00
More than 75 but 125 or less	750.00
More than 125 but 200 or less	1225.00
More than 200 but 350 or less	1650.00
More than 350	2000.00

Write a program to input the water consumed annually in gallons and output the tax payable.

**Ans.**

```

import java.util.*;
class Q37
{
    static void main()
    {
        int g;
        double tax;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the water consumed annually in gallons:");
        g=sc.nextInt();
        if (g<=45)

```

```

tax=0;
else if(g>45 && g<=75)
tax=g/100f*475;
else if(g>75 && g<=125)
tax=g/100f*750;
else if(g>125 && g<=200)
tax=g/100f*1225;
else if(g>200 && g<=350)
tax=g/100f*1650;
else
tax=g/100f*2000;
}
}

```

**38. Write a program to input the length of the 3 sides of a triangle (say a, b and c) and calculate the area depending upon the following criteria:**

**i. If the sum of any two sides of the triangle is lesser than the third side then print “A triangle cannot be formed”**

**ii. If a=b=c then print “It is an equilateral triangle” and calculate the area as**

$$area = \frac{\sqrt{3}}{2} a^2$$

**iii. If the sum of the squares of any two sides is equal to the square of the third side then print “It is a right-angled triangle” and calculate the area as**

$$area = \frac{1}{2} * \text{product of the perpendicular sides}$$

**iv. otherwise calculate area as area= where  $\sqrt{s(s-a)(s-b)(s-c)}$ .**

$$s = \frac{a+b+c}{2}$$

**Ans.**

```

import java.util.*;
class Q38
{
static void main()
{
int a,b,c;
double ar,s;

```

```

Scanner sc=new Scanner(System.in);
System.out.println("Enter the 3 sides of the triangle:");
a=sc.nextInt();
b=sc.nextInt();
c=sc.nextInt();
if (a<b+c && b<c+a && c<a+b)
{
if (a==b && b==c)
ar=Math.sqrt(3)/2*a*a;
else if(a*a==b*b +c*c)
ar=1/2.0*(b*c);
else if(b*b==a*a +c*c)
ar=1/2.0*(a*c);
else if(c*c==b*b +a*a)
ar=1/2.0*(b*a);
else
{
s=(a+b+c)/2.0;
ar=Math.sqrt(s*(s-a)*(s-b)*(s-c));
}
System.out.println("Area="+ar);
}
else
System.out.println("A triangle cannot be formed");
}
}

```

39. The roots (say  $x_1$  and  $x_2$ ) of a quadratic equation  $ax^2+bx+c=0$  can be determined by its discriminant ( $d$ ), where  $d=b^2-4ac$ . Depending upon its value the nature of the roots and the value of the can be determined.

a. If  $d=0$  then the roots are Real and Equal and the roots can be found using the formula:

$$\frac{-b}{2a}$$

b. If  $d>0$  then the roots are Real and Distinct and the roots can be found using.

$$\frac{-b \pm \sqrt{d}}{2a} \text{ and } \frac{-b - \sqrt{d}}{2a}$$

c. If  $d < 0$  then the roots are Imaginary and the roots can be found.

$$x_1 = \frac{-d}{2a}$$

$$x_2 = \frac{\sqrt{-d}}{2 * a}$$

Write a program to input a,b and c the coefficient of the quadratic equation and find its roots and/or nature.

Ans.

```
import java.util.*;
class Q39
{
static void main()
    {
    int a,b,c;
    double d,x1,x2;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the coefficients of the quadratic equation:");
    a=sc.nextInt();
    b=sc.nextInt();
    c=sc.nextInt();
    d=b*b-4*a*c;
    if (d<0)
    System.out.println("Roots are imaginary");
    else if(d==0)
    {
    System.out.println("Roots are real and equal");
    x1=x2=(double)-b/(2*a);
    System.out.println("Root 1:"+x1);
    System.out.println("Root 1:"+x2);
    }
    else
    {
    x1=(-b+Math.sqrt(d))/(2*a);
    x2=(-b-Math.sqrt(d))/(2*a);
    System.out.println("Root 1:"+x1);
    System.out.println("Root 1:"+x2);
    }
    }
}
```

## Chapter 9

# Iterative Constructs in Java



Tick (✓) the correct option.

1. A type of loop that is usually used when the number of iteration is known.
- a. for
  - b. while
  - c. do-while
  - d. None of these

**Ans.** a. for

2. A type of loop when the loop executes atleast once even if the test expression evaluates to 'true'.
- a. for
  - b. while
  - c. do-while
  - d. None of these

**Ans.** c. do-while

3. In while and do-while loops, a statement causes control to be transferred directly to the conditional expression that controls the loop.
- a. break
  - b. pause
  - c. start
  - d. continue

**Ans.** d. continue

4. By using you can force immediate termination of loop, bypassing the conditional expression and any remaining code in the body of the loop.
- a. switch
  - b. break
  - c. continue
  - d. default

**Ans.** b. break

5. What is an infinite loop?
- a. A loop that functions infinitely well
  - b. A loop that runs forever
  - c. A loop that never starts
  - d. A loop that will never function

**Ans.** b. A loop that runs forever

6. What is the value of k after the following code fragment?

```
int k = 0;
int n = 12
while (k < n)
{
```

```
        k = k + 1;
    }
```

- a. 0                      b. 11                      c. 12                      d. Unknown

**Ans. c. 12**

7. Which of these jump statements can skip processing remainder of code in its body for a particular iteration?
- a. break    b. return  
c. exit    d. continue

**Ans. d. continue**

8. Which looping process checks the test condition at the end of the loop?
- a. for    b. while  
c. do-while                                        d. no looping process checks the test condition at the end

**Ans. c. do-while**

9. What's wrong? for (int k = 2, k <= 12, k++)
- a. the increment should always be ++k  
b. the variable must always be the letter i when using a for loop  
c. there should be a semicolon at the end of the statement  
d. the commas should be semicolons

**Ans. d. the commas should be semicolons**

10. What's wrong? while( (i < 10) && (i > 24))
- a. the logical operator && cannot be used in a test condition  
b. the while loop is an exit-condition loop  
c. the test condition is always false  
d. the test condition is always true

**Ans. c. the test condition is always false**

## Section A

**Answer the following questions.**

1. **Explain the term loop with an example.**

**Ans.** A *loop* is a sequence of instructions that is continually repeated until a certain condition is reached.

The following program illustrates the use of **for** loop.

```
public class Loops
```

```

{
static void NaturalNumbers()
    {
    int i;
    for(i=1 ; i<=10 ; i++)
    {
    System.out.println(i);
    }
    }
}

```

Now let us learn how the components of the loop gets executed.

Step 1:

At the beginning the initialization expression is executed *i.e.* `i=1` which assigns 1 to `i`.

Step 2:

Then the test expression is checked *i.e.* `i<=10` which checks whether the value of `i` is less than or equal to 10.

Step 3:

Since the test expression is **true** it enters the body of the loop *i.e.* `System.out.println(i);` is executed which prints the current value of `i` in the next line.

Step 4:

After executing the loop-body, the update expression *i.e.* `i++` is executed which increments the value of `i`. (The value of `i` becomes 2 after the execution of `i++`, since initially `i` was 1).

Step 5:

After the update expression is executed, the test-expression is again evaluated. If it is **true** the sequence is repeated from step number 3, otherwise the loop terminates.

## 2. Name the Elements that Control a Loop.

**Ans.**

Every loop has elements that control its execution. These elements can be grouped under the following in general:

- 1. Initialization Expression(s).** Before entering a loop, the variable that controls the execution of it is called initialization. The initialization of a variable may be inside a loop or outside it. Whatever it may be it is executed only once before the iteration starts.
- 2. Test Expression.** The *test-expression (also called test-condition)* is a condition, which decides whether the *loop body* will be executed or not. Till the *test expression* evaluates to true, the *loop body* gets executed otherwise the loop is terminated.

- 3. Update Expression(s).** The *update expression(s)* change the value(s) of the control variable(s). The *update expression(s)* is usually made to be executed; before the next iteration.
- 4. The Body of the Loop.** The statements that are executed repeatedly (as long as the, *test expression* is *true*) form the body of the loop.

**3. What are the three types of looping statement available in Java?**

**Ans.** The three types of looping statements are: for, while and do-while.

**4. Classify loops depending upon the priority in which a test-expression is evaluated.**

**Ans.**

Depending upon the way a test-expression is evaluated in a loop it can be classified under the following:

Entry controlled loop

Exit control loop

In an entry controlled loop, the *test expression* is evaluated before entering into a loop whereas in an *exit controlled loop*, the *test expression* is evaluated before exiting from the loop.

**5. Give the general syntax of a for-loop.**

**Ans.** The syntax of the **for** loop statement is

```
for (initialization expression(s); test expression; update expression(s))
{
    body of the loop;
}
```

**6. Give an example to show how multiple initialization and updation are performed in a for-loop.**

**Ans.**

```
int i, j;
for(i=1, j=10 ; i<=10 ; j-- , i++)
{
    System.out.println("i=" + i+ " "+ "j="+j);
}
```

**7. What is an empty for-loop. Give an example.**

**Ans.**

If the loop doesn't contain a body, it is called an empty loop. For example,

```
for(i=1;i<=100;i++);
```

Notice the semicolon at the end of the parenthesis, indicating that the **for** loop does not have body and therefore no statement is repeatedly executed again and again.



**8. State one similarity and one difference between while and do while loop.**

**Ans. Similarity:** Both the loops are generally used if the number of iterations are not known.

**Difference:** The while loop is an entry controlled loop and the do-while is an exit controlled loop.

**9. State one similarity and one difference between while and for loop.**

**Ans. Similarity:** Both are entry controlled loops.

**Difference:** Initialisation, test expression and updation can be written at the beginning of a for-loop, whereas in a while loop, the initialisation is written outside the loop, and the updation inside the body of the loop.

**10. What is meant by an infinite loop ? Give one example.**

**Ans.** A loop whose test expression always results to true is an infinite loop. For example,

```
for(int i=1;i!=0;i++)  
{  
}
```

**11. 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)  
{  
body-of-the-loop;  
}
```

**12. 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 **do-while** loop is:

```
do  
{  
statement;  
}while (condition);
```

To create an infinite you need to make the condition always result to true. For example,

```
do  
{  
System.out.println("Hello");  
}while (true);
```

**13. Compare loops according to its usage.**

**Ans.** The **for** loop is appropriate when you know in advance how many times the loop will be executed, *i.e.*, the first and the last limit is already known. The other two loops **while** and **do-while** loops are more suitable in situations when it is not known in advance when the loop will terminate. The **while** should be preferred when you may not want to execute the *loop-body* even once (in case the *test condition* is **false**), and the **do-while** loop should be preferred when you're sure you want to execute the *loop-body* at least once.

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

**Ans.** A loop within another loop is called nested loop. For example,

```
int x , y;  
for(x=1;x<=3;x++)  
{  
    for(y=1; y<=3 ; y++)  
    {  
        System.out.println(x+ " " +y);  
    }  
}
```

**15. Name two jump statements and their use.**

**Ans.** The 'break' statement is used to exit from a loop or switch block.

The 'continue' statement is used to skip the remaining statements in a loop and continue with the next iteration.

**B. Answer as directed:**

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

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

```
x=5;           y=50;
while(x<=y)
{
y=y/x;
System.out.println(y);
}
```

Ans.

x	y	x<=y	y=y/x
5	50	true	10
5	10	true	2
5	2	false	-

The loop executes twice.

3. Convert the following segment into equivalence for loop.

```
int i=0;
while(i<=20)
{
System.out.print(i+ " ");
i++;
}
```

Ans.

```
int i;
for(i=0;i<=20;i++)
{
System.out.print(i+ " ");
}
```

4. 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=15

5. Give the output of the following program:

```
class Output1  
{  
    static void find()  
    {  
        int n=1234, d ,s=0;  
        while(n!=0)  
        {  
            d=n%10;  
            s=s*10+d;  
            n=n/10;  
        }  
        System.out.println(s);  
    }  
}
```

**Ans.**

4321

6. Give the output of the following program:

```
class Output2  
{  
    static void find()  
    {  
        int n=1234, d ,s=0,c=0;  
        while(n!=0)
```

```

        {
            d=n%10;
            s=s+d*(int)Math.pow(10,c++);
            n=n/10;
        }
        System.out.println(s);
    }
}

```

**Ans.** 1234

## SECTION B

**Write programs for the following.**

1. Write a program to find the sum of 1st 10 odd natural numbers.

**Ans.**

```

class Q1
{
    static void main()
    {
        int i,s=0;
        for(i=1;i<=19;i+=2)
        {
            s=s+i;
        }
        System.out.println(s);
    }
}

```

2. Write a program to find the sum of 1st 10 even natural numbers.

**Ans.**

```

class Q2
{
    static void main()
    {
        int i,s=0;
        for(i=2;i<=20;i+=2)

```

```

    {
    s=s+i;
    }
    System.out.println(s);
    }
}

```

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

**Ans.**

```

class Q3
{
static void main()
    {
    int i,s=0;
    for(i=100;i<=998;i+=2)
    {
    s=s+i;
    }
    System.out.println(s);
    }
}

```

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

**Ans.**

```

class Q4
{
static void main()
    {
    int i,s=0;
    for(i=101;i<=999;i+=2)
    {
    if (i%5==0)
    s=s+i;
    }
    System.out.println(s);
    }
}

```

5. 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  $1 \times 2 \times 3 \times 4 \times 5 = 120$ .

**Ans.**

```
import java.util.*;
class Q5
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,f=1,n;
System.out.println("Enter a number:");
n=sc.nextInt();
for(i=1;i<=n;i++)
    {
f=f*i;
    }
System.out.println("Factorial="+f);
    }
}
```

6. Write a program to input an integer and print its factors.

**For Example,**

**INPUT: Enter an integer:12**

**OUTPUT: Factors: 1 2 3 4 6 12**

**Ans.**

```
import java.util.*;
class Q6
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,n;
System.out.println("Enter an integer:");
n=sc.nextInt();
System.out.println("Factors:");
for(i=1;i<=n;i++)
    {
```

```

        if (n%i==0)
        System.out.print(i+" ");
    }
}
}

```

**7. Write a program to input an integer and count the number of factors.**

**Ans.**

```

import java.util.*;
class Q7
{
static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,n,c=0;
        System.out.println("Enter an integer:");
        n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            if (n%i==0)
            c++;
        }
        System.out.println("No. of Factors:"+c);
    }
}

```

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

**Ans.**

```

import java.util.*;
class Q8
{
static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,n,c=0;
        System.out.println("Enter an integer:");
        n=sc.nextInt();
    }
}

```



```

for(i=1;i<=n;i++)
{
if (n%i==0)
c++;
}
if(c==2)
System.out.println("Prime number");
else
System.out.println("Not a Prime number");
}
}

```

**9. Write a program to input 10 integers and find their sum.**

**Ans.**

```

import java.util.*;
class Q9
{
static void main()
{
Scanner sc=new Scanner(System.in);
int i,n,s=0;
System.out.println("Enter 10 integers:");
for(i=1;i<=10;i++)
{
n=sc.nextInt();
s=s+n;
}
System.out.println("Sum of the numbers="+s);
}
}

```

**10. Write a program to input 10 integers and find the sum of even numbers only.**

**Ans.**

```

import java.util.*;
class Q10
{

```

```

static void main()
{
    Scanner sc=new Scanner(System.in);
    int i,n,s=0;
    System.out.println("Enter 10 integers:");
    for(i=1;i<=10;i++)
    {
        n=sc.nextInt();
        if(n%2==0)
            s=s+n;
    }
    System.out.println("Sum of even numbers="+s);
}

```

- 11. 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 Q11
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,n,s2=0,s3=0;
        System.out.println("Enter 10 integers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            if(n>=10 && n<=99)
                s2=s2+n;
            if(n>=100 && n<=999)
                s3=s3+n;
        }
        System.out.println("Sum of 2 digit numbers="+s2);
        System.out.println("Sum of 3 digit numbers="+s3);
    }
}

```

**12. Write a program to input 10 integers and display the largest integer.**

**Ans.**

```
import java.util.*;
class Q12
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,n,c=0,l=0;
System.out.println("Enter 10 integers:");
for(i=1;i<=10;i++)
    {
n=sc.nextInt();
c++;
if(c==1)
l=n;
if(n>l)
l=n;
    }
System.out.println("Largest number="+l);
    }
}
```

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

**Ans.**

```
import java.util.*;
class Q13
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,n,c=0,l=0,s=0;
System.out.println("Enter 10 integers:");
for(i=1;i<=10;i++)
    {
```

```

n=sc.nextInt();
c++;
if(c==1)
s=l=n;
if(n>l)
l=n;
if(n<s)
s=n;
}
System.out.println("Largest number="+l);
System.out.println("Smallest number="+s);
}
}

```

- 14. Write a program to input 10 integers and display the largest even integer. In case there is no even integer, it should print "No even integer found".**

**Ans.**

```

import java.util.*;
class Q14
{
static void main()
{
Scanner sc=new Scanner(System.in);
int i,n,c=0,l=0;
System.out.println("Enter 10 integers:");
for(i=1;i<=10;i++)
{
n=sc.nextInt();
if(n%2==0)
{
c++;
if(c==1)
l=n;
if(n>l)
l=n;
}
}
}
}

```

```

    }
    if(c>0)
        System.out.println("Largest even number="+l);
    else
        System.out.println("No even integer found");
    }
}

```

**15. Write a program to input 10 integers and display the largest even and smallest odd integer.**

**Ans.**

```

import java.util.*;
class Q15
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,n,c=0,l=0,p=0,s=0;
        System.out.println("Enter 10 integers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            if(n%2==0)
            {
                c++;
                if(c==1)
                    l=n;
                if(n>l)
                    l=n;
            }
            else
            {
                p++;
                if(p==1)
                    s=n;
                if(n<s)
                    s=n;
            }
        }
    }
}

```

```

    }
    if(c>0)
    System.out.println("Largest even number="+l);
    else
    System.out.println("No even integer found");
    if(p>0)
    System.out.println("Smallest odd number="+s);
    else
    System.out.println("No odd integer found");
    }
}

```

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

**Ans.**

```

import java.util.*;
class Q16
{
static void main()
    {
    Scanner sc=new Scanner(System.in);
    int i,n,f=0;
    System.out.println("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");
    }
}

```

17. 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 Q17
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,n,f=0,m=0;
System.out.println("Enter 10 integers:");
for(i=1;i<=10;i++)
    {
n=sc.nextInt();
if(i==1)
m=n;
if(n!=m)
f=1;
    }
if(f==0)
System.out.println("All numbers are same");
else
System.out.println("All numbers are not same");
    }
}
```

18. 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 Q18
{
static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,n,f=0,p=0;
        System.out.println("Enter 10 integers:");
        for(i=1;i<=10;i++)
        {
            n=sc.nextInt();
            if(i>1)
            {
                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");
        }
}
```

**19 Write a program to calculate and print the sum of odd numbers and the sum of even numbers for the first n natural numbers. The integer n is to be entered by the user.**

**Ans.**

```
import java.util.*;
class Q19
{
static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,n,se=0,so=0;
```



```

System.out.println("Enter a number:");
n=sc.nextInt();
for(i=1;i<=n;i++)
{
if(n%2==0)
se+=n;
else
so+=n;
}
System.out.println("Sum of even numbers="+se);
System.out.println("Sum of odd numbers="+so);
}
}

```

20. Write programs to find the sum of each of the following series:

i.  $s=1*2+2*3+3*4+\dots+9*10$

ii.  $s=1*2+3*4+5*6+\dots+9*10$

iii.  $s=1*3+2*4+3*5+\dots+9*11$

iv.  $s=1-2+3-4+5-6+\dots+19-20$

v.  $s=1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\dots+\frac{1}{10}$

vi.  $s=\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\frac{4}{5}+\dots+\frac{9}{10}$

vii.  $s=1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\frac{1}{5}+\dots-\frac{1}{10}$

Ans.

```

class Q20
{
static void i()
{
int i,s=0;
for(i=1;i<=9;i++)
{
s=s+i*(i+1);
}
System.out.println("Sum =" +s);
}
}

```

```

}
static void ii()
{
int i,s=0;
for(i=1;i<=9;i+=2)
{
s=s+i*(i+1);
}
System.out.println("Sum =" +s);
}
static void iii()
{
int i,s=0;
for(i=1;i<=9;i++)
{
s=s+i*(i+2);
}
System.out.println("Sum =" +s);
}
static void iv()
{
int i,s=0;
for(i=1;i<=20;i++)
{
if(i%2==0)
s=s-i;
else
s=s+i;
}
System.out.println("Sum =" +s);
}
static void v()
{
int i;
float s=0;

```

```

for(i=1;i<=10;i++)
{
s=s+(float)1/i;
}
System.out.println("Sum =" +s);
}
static void vi()
{
int i;
float s=0;
for(i=1;i<=9;i++)
{
s=s+(float)i/(i+1);
}
System.out.println("Sum =" +s);
}
static void vii()
{
int i;
float s=0;
for(i=1;i<=10;i++)
{
if(i==2 || i==10)
s=s-(float)i/(i+1);
else
s=s+(float)i/(i+1);
}
System.out.println("Sum =" +s);
}
}

```

21. Write a program to input two integers say x and n and find the sum of the following series:

- i.  $x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \frac{x^5}{5} + \frac{x^6}{6} + \frac{x^7}{7} + \dots + \frac{x^n}{n}$
- ii.  $\frac{x}{2} + \frac{x^2}{3} + \frac{x^3}{4} + \frac{x^4}{5} + \frac{x^5}{6} + \frac{x^6}{7} + \dots + \frac{x^n}{n+1}$
- iii.  $x^2 + \frac{x^3}{2} + \frac{x^4}{3} + \frac{x^5}{4} + \frac{x^6}{5} + \frac{x^7}{6} + \dots + \frac{x^{n+1}}{n}$

$$\text{vi. } x^2 + \frac{x^3}{2} + \frac{x^4}{3} + \frac{x^5}{4} + \frac{x^6}{5} + \frac{x^7}{6} + \dots + \frac{x^{n+1}}{n}$$

$$\text{v. } x - \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \frac{x^5}{5} + \frac{x^6}{6} + \frac{x^7}{7} - \dots \pm \frac{x^n}{n}$$

**Ans.**

```
import java.util.*;
class Q21
{
static void i()
    {
    int i,x,n;
    double s=0;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the values of x and n:");
    x=sc.nextInt();
    n=sc.nextInt();
    for(i=1;i<=n;i++)
    {
    s=s+Math.pow(x,i)/i;;
    }
    System.out.println("Sum =" +s);
    }
static void ii()
    {
    int i,x,n;
    double s=0;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the values of x and n:");
    x=sc.nextInt();
    n=sc.nextInt();
    for(i=1;i<=n;i++)
    {
    s=s+Math.pow(x,i)/(i+1);;
    }
    System.out.println("Sum =" +s);
    }
```

```

}
static void iii()
{
int i,x,n;
double s=0;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the values of x and n:");
x=sc.nextInt();
n=sc.nextInt();
for(i=1;i<=n;i++)
{
s=s+Math.pow(x,i+1)/i;
}
System.out.println("Sum =" +s);
}
static void iv()
{
int i,x,n;
double s=0;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the values of x and n:");
x=sc.nextInt();
n=sc.nextInt();
for(i=1;i<=n;i++)
{
s=s+Math.pow(x,i+1)/i;
}
System.out.println("Sum =" +s);
}
static void v()
{
int i,x,n;
double s=0;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the values of x and n:");

```

```

    x=sc.nextInt();
    n=sc.nextInt();
    for(i=1;i<=n;i++)
    {
        s=s+Math.pow(x,i)/i;;
    }
    System.out.println("Sum =" +s);
}
}

```

22. Write a program to find the sum of 1st 10 numbers of Fibonacci series i.e. 1,1,2,3,5,8,13.... Fibonacci series is such a series which starting from 1 and 1, and subsequent numbers are the sum of the previous two numbers.

**Ans.**

```

class Q22
{
    static void main()
    {
        int i,f=1,s=0,sum=0,t;
        for(i=1;i<=10;i++)
        {
            t=f+s;
            sum+=t;
            f=s;
            s=t;
        }
        System.out.print("Sum="+sum);
    }
}

```

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 Q23
{

```

```

static void main()
{
    int i,f=1,s=0,t;
    for(i=1;i<=15;i++)
    {
        t=f+s*2;
        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 Q24
{
    static void main()
    {
        int i,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 Q25
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,n,s=0;
System.out.println("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 Q26
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,a,b,sa=0,sb=0;
```



```

System.out.println("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 Amicable pair");
}
}

```

**27. Write a program to pass an integer as argument and find the sum of its digits.**

**Ans.**

```

class Q27
{
static void main(int a)
{
int i,d,s=0;
for(i=a;i>0;i=i/10)
{
d=i%10;
s=s+d;
}
System.out.println("Sum="+s);
}
}

```

**28. Write a program to pass an integer as argument and find the sum of odd digits and even digits separately.**

**Ans.**

```
class Q28
{
static void main(int a)
    {
    int i,d,se=0,so=0;
    for(i=a;i>0;i=i/10)
    {
    d=i%10;
    if(d%2==0)
    se=se+d;
    else
    so=so+d;
    }
    System.out.println("Sum of even digits="+se);
    System.out.println("Sum of odd digits="+so);
    }
}
```

**29. Write a program to pass an integer as argument and find the average of its digits.**

**Ans.**

```
class Q29
{
static void main(int a)
    {
    int i,d,s=0,c=0;
    float avg;
    for(i=a;i>0;i=i/10)
    {
    d=i%10;
    s+=d;
    c++;
    }
    avg=(float)s/c;
    System.out.println("Average="+avg);
    }
}
```

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

**Ans.**

```
class Q30
{
static void main(int a)
    {
    int i,d,l=0,s=0,c=0;
    for(i=a;i>0;i=i/10)
    {
    d=i%10;
    c++;
    if(c==1)
    l=s=d;
    if(d>l)
    l=d;
    if(d<s)
    s=d;
    }
    System.out.println("Largest="+l);
    System.out.println("Smallest="+s);
    }
}
```

**31. 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.**

```
class Q31
{
static void main(int a)
    {
    int i,d,s=0,c=0;
    for(i=a;i>0;i=i/10)
    {
    d=i%10;
    if(d%2!=0)
```

```

        s=s+d*(int)Math.pow(10,c++);
    }
    System.out.println("New number="+s);
}
}

```

- 32. Write a program to pass an integer as argument and check whether all digits in it are even digit or not.**

**Ans.**

```

class Q32
{
static void main(int a)
    {
    int i,d,f=0;
    for(i=a;i>0;i=i/10)
    {
    d=i%10;
    if(d%2!=0)
    f=1;
    }
    if(f==0)
    System.out.println("All are even digits");
    else
    System.out.println("All are not even digits");
    }
}

```

- 33. Write a program to pass an integer as argument and check whether the digits are in ascending order or not.**

**Ans.**

```

class Q33
{
static void main(int a)
    {
    int i,d,d1,f=0;
    for(i=a;i>9;i=i/10)
    {

```

```

        d=i%10;
        d1=(i/10)%10;
        if(d<d1)
            f=1;
        }
        if(f==0)
            System.out.println("Digits are in ascending order");
        else
            System.out.println("Digits are not in ascending order");
        }
    }
}

```

34. Write a program to pass an integer as argument and check whether it is Armstrong number or not. Numbers whose sum of the cube of its digit is equal to the number itself is called Armstrong numbers. Example  $15^3=1^3+5^3+3^3$ .

**Ans.**

```

class Q34
{
static void main(int a)
    {
        int i,d,s=0;
        for(i=a;i>9;i=i/10)
        {
            d=i%10;
            s=s+d*d*d;
        }
        if(s==a)
            System.out.println("Armstrong Number");
        else
            System.out.println("Not an Armstrong Number");
        }
    }
}

```

35. Write a program to input an integer and check whether it is an automorphic, trimorphic or tri-automorphic number or not. A number  $n$  is said to be automorphic, if its square ends in  $n$ . For instance 5 is automorphic, because  $5^2=25$ , which ends in 5, 25 is automorphic, because  $25^2=625$ , which ends in 25. A number  $n$  is called trimorphic if  $n^3$  ends in  $n$ . For example  $49^3=117649$ , is trimorphic. A number  $n$  is called tri-automorphic if  $3n^2$  ends in  $n$ ; for example  $667$  is tri-automorphic because  $3 \times 667^2=1334667$ , ends in 667.

**Ans.**

```
import java.util.*;
class Q35
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,d,a,c=0;
System.out.println("Enter an integer:");
a=sc.nextInt();
for(i=a;i>0;i=i/10)
    {
c++;
    }
if((a*a)%((int)Math.pow(10,c))==a)
System.out.println("Automorphic Number");
else if((a*a*a)%((int)Math.pow(10,c))==a)
System.out.println("Trimorphic Number");
else if((3*a*a)%((int)Math.pow(10,c))==a)
System.out.println("Tri-Automorphic Number");
else
System.out.println("Not any type of Automorphic Number");
    }
}
```

- 36. 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 Q36
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int i,d,a,s=0;
System.out.println("Enter an integer:");
a=sc.nextInt();
for(i=a;i>0;i=i/10)
```

```

    {
    d=i%10;
    s+=d;
    }
    if(a%s==0)
    System.out.println("Harshad Number");
    else
    System.out.println("Not a Harshad Number");
    }
}

```

37. 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 ( $9^2 = 81, 8 + 1 = 9$ ), 45 ( $45^2 = 2025, 20 + 25 = 45$ ), and 297 ( $297^2 = 88209, 88 + 209 = 297$ ).

**Ans.**

```

import java.util.*;
class Q37
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,d=0,a,s=0,f,l;
        System.out.println("Enter an integer:");
        a=sc.nextInt();
        for(i=a;i>0;i=i/10)
        {
            d++;
        }
        s=a*a;
        f=s/(int)Math.pow(10,d);
        l=s%(int)Math.pow(10,d);
        if(f+l==a)
        System.out.println("Kaprekar Number");
        else
        System.out.println("Not a Kaprekar Number");
        }
}

```

**38. 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 Q38
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,a,b,l=0;
        System.out.println("Enter 2 integer:");
        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);
    }
}
```

**39. 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 Q39
{
```



```

static void main()
{
Scanner sc=new Scanner(System.in);
int i,a,b,h=0;
System.out.println("Enter 2 integers:");
a=sc.nextInt();
b=sc.nextInt();
for(i=a;i>=1;i--)
{
if(a%i==0 && b%i==0)
{
h=i;
break;
}
}
System.out.println("H.C.F.="+h);
}
}

```

**40. 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 Q40
{
static void main()
{
Scanner sc=new Scanner(System.in);
int i,n,r=0,d,ch,s=0;
System.out.println("Enter a number:");
n=sc.nextInt();
System.out.println("MENU");

```

```

System.out.println("1.Palindrome Number");
System.out.println("2.Perfect Number");
System.out.println("Enter your choice:");
ch=sc.nextInt();
switch(ch)
{
case 1:
for(i=n;i>0;i/=10)
{
d=i%10;
r=r*10+d;
}
if(r==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=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");
}
}
}

```

41. 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 BUZZ number is the number which either ends with 7 or divisible by 7.
  - 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 Q41
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int ch,n,r=0,a,b;
System.out.println("MENU");
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%7==0 || n%10==7)
System.out.println("Buzz Number");
else
System.out.println("Not a Buzz Number");
break;
case 2:
System.out.println("Enter 2 numbers:");
a=sc.nextInt();
b=sc.nextInt();
for(r=a%b;r>0;r=a%b)
{
a=b;
b=r;
```

```

    }
    System.out.println("GCD="+b);
    break;
    default:
    System.out.println("Invalid choice");
    }
}

```

42. 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.

- 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 Q42
{
static void main()
    {
    Scanner sc=new Scanner(System.in);
    int ch,i,c=0,n;
    System.out.println("Enter a number:");
    n=sc.nextInt();
    System.out.println("MENU");
    System.out.println("1.Prime Number");
    System.out.println("2.Automorphic Number");
    System.out.println("Enter your choice:");
    ch=sc.nextInt();
    switch(ch)
    {
    case 1:
    for(i=1;i<=n;i++)
    {
    if(n%i==0)
    c++;
    }
    }
    }

```

```

    }
    if(c==2)
        System.out.println("Prime Number");
    else
        System.out.println("Not a Prime Number");
    break;
    case 2:
        for(i=n;i>0;i=i/10)
        {
            c++;
        }
        if((n*n)%(int)Math.pow(10,c)==n)
            System.out.println("Automorphic Number");
        else
            System.out.println("Not an Automorphic Number");
        break;
    default:
        System.out.println("Invalid choice");
    }
}
}
}

```

43. Write a program to input 10 integers and print the second largest number. Assume that there is at least one second largest number in the given set of integers.

For Example,

**INPUT: Enter 10 integers:**

**12 35 46 22 34 56 78 89 23 21**

**OUTPUT: Second Largest Integer: 78**

**Ans.**

```

import java.util.*;
class Q43
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int l=0,sl=0,i,c=0,n;
        System.out.println("Enter 10 numbers:");
    }
}

```

```

    for(i=1;i<=10;i++)
    {
        n=sc.nextInt();
        if(i==1)
            l=n;
        else if(n>l)
        {
            sl=l;
            l=n;
        }
        else
        {
            if(l!=0)
            {
                if(sl==0)
                {
                    sl=n;
                }
                if(n>sl)
                {
                    sl=n;
                }
            }
        }
        System.out.println("Second Largest="+sl);
    }
}

```

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

**$S=1+(1+2)+(1+2+3)+(1+2+3+4)+(1+2+3+4+5)+\dots+(1+2+3+4+\dots+10)$**

**(Please note that no nested loop is to be used)**

**Ans.**

```

class Q44
{
    static void main()
    {
        int i,s=0,p=0;
        for(i=1;i<=10;i++)
        {
            p=p+i;

```

```

        s=s+p;
    }
    System.out.println("Sum="+s);
}
}

```

45. Write a program to print the sum of negative numbers, sum of positive even numbers and sum of positive odd numbers from a list of numbers (N) entered by the user. The list terminates when the user enters a zero.

**Ans.**

```

import java.util.*;
class Q45
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int i,n,se=0,so=0;
        for(;true;)
        {
            System.out.println("Enter a number:");
            n=sc.nextInt();
            if(n==0)
                break;
            if(n%2==0 && n>0)
                se+=n;
            else if(n%2==1 && n>0)
                so+=n;
        }
        System.out.println("Sum of positive even no.s="+se);
        System.out.println("Sum of positive odd no.s="+so);
    }
}

```

46. A game of dice is to be simulated for two players, each player gets a chance to throw his dice, and the value is added to his points, this process continues alternately until for the player whose added points equals to 20 and is declared the winner. Write a program to simulate this process using the random( ) function.

**Ans.**

```
class Q46
{
static void main()
    {
    int i,r1,r2,s1=0,s2=0;
    for(i=1;s1<20 && s2<20;i++)
    {
    r1=1+(int)(Math.random()*20);
    s1+=r1;
    if(s1<20)
    {
    r2=1+(int)(Math.random()*20);
    s2+=r2;
    }
    }
    if(s1>s2)
    System.out.println("Winner is player 1");
    else
    System.out.println("Winner is player 2");
    }
}
```

**47. Write a program to calculate and print the values of:-**

$$Z = \frac{x^2 + y^2}{x + y}$$

**Where, x ranges from 0 to 50 and y is to be taken as input.**

**Ans.**

```
import java.util.*;
class Q47
{
static void main()
    {
    Scanner sc=new Scanner(System.in);
    int x,y,z;
```



```

    for(x=0;x<=50;x++)
    {
        System.out.println("Enter the value of y:");
        y=sc.nextInt();
        z=(x*x+y*y)/(x+y);
        System.out.println("z="+z);
    }
}

```

48. 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) =  $\frac{x}{2} + \frac{x}{5} + \frac{x}{8} + \frac{x}{11} + \dots + \frac{x}{20}$

Ans.

```

import java.util.*;
class Q48
{
    static void a()
    {
        int s=0,i;
        for(i=1;i<=10;i++)
        {
            if(i%2==0)
            s=s-2*i;
            else
            s=s+2*i;
        }
        System.out.println("Sum="+s);
    }
    static void b()
    {
        Scanner sc=new Scanner(System.in);
        int x,i;
        float s=0;
        System.out.println("Enter the value of x:");
        x=sc.nextInt();
        for(i=2;i<=20;i+=3)
    }
}

```

```

    {
    s=s+(float)x/i;
    }
    System.out.println("Sum="+s);
    }
}

```

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

$$S = \frac{a}{2} + \frac{a}{3} + \frac{a}{4} + \dots + \frac{a}{n}$$

Ans.

```

import java.util.*;
class Q49
{
static void main()
    {
    Scanner sc=new Scanner(System.in);
    int a,n,i;
    float s=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=2;i<=n;i++)
    {
    s=s+(float)a/i;
    }
    System.out.println("Sum="+s);
    }
}

```

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

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

Ans.

```

import java.util.*;
class Q50
{
static void main()
    {

```

```

Scanner sc=new Scanner(System.in);
int a,n,i,c=1;
float s=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++)
{
s=s+(float)(a+c)/(c+1);
c+=2;
}
System.out.println("Sum="+s);
}

```

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

$$s = \frac{1+2}{2*3} + \frac{2+3}{3*4} + \frac{3+4}{4*5} + \frac{4+5}{5*6} \dots \text{upto } n \text{ terms}$$

Ans.

```

import java.util.*;
class Q51
{
static void main()
{
Scanner sc=new Scanner(System.in);
int n,i;
float s=0;
System.out.println("Enter the value of n:");
n=sc.nextInt();
for(i=1;i<=n;i++)
{
s=s+(float)(i+(i+1))/((i+1)*(i+2));
}
System.out.println("Sum="+s);
}
}

```

52. An interesting method of multiplication of integers (not very large numbers) is as follows. The method can be illustrated by the following example. If the numbers 17 and 19 are to be multiplied, they are put at the top of two columns as shown below:

17

19

The numbers at the left hand side is successively divided by 2 (integer division) while the other is successively multiplied by 2. The results are written one below the other in their respective columns. The process is repeated till the column containing the division results reached 1. At this stage all the numbers in the right hand column are struck off where numbers corresponding to it on the left hand column is even.

17	19
8 (even)	38
4 (even)	76
2 (even)	<del>152</del>
1	304

Now the remaining numbers on the right hand side are added:  $19+304=323$  which is equal to the product of 17 and 19.

Write a program to input two 2-digit numbers and find their product using the above technique.

**Ans.**

```
import java.util.*;
class Q52
{
static void main()
    {
Scanner sc=new Scanner(System.in);
int a,b,i,s=0;
System.out.println("Enter 2 numbers:");
a=sc.nextInt();
b=sc.nextInt();
for(i=a;i>=1;i=i/2)
    {
if (i%2==1)
s=s+b;
b=b*2;
}
System.out.println("Product="+s);
}
}
```

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 Q53
{
    static void main()
    {
        Scanner sc=new Scanner(System.in);
        int n,i,f=2;
        System.out.println("Enter a number:");
        n=sc.nextInt();
        for(i=n;i>1;)
        {
            if(i%f==0)
            {
                System.out.println(f);
                i=i/f;
            }
            else
                f++;
        }
    }
}
```

# Chapter 10

## Nested For Loops



Tick (✓) the correct option.

1. In a group of nested loops, which loop is executed the most number of times?
  - a. the outermost loop
  - b. the innermost loop
  - c. all loops are executed the same number of times
  - d. cannot be determined without knowing the size of the loops

**Ans.** b. the innermost loop

2. Examine the following code. How many times will the nested loop run?

```
for (inti = 1; i <= 3; i++) {  
  for (int j = 1; j <= i; j++) {  
  }  
}
```

- a. 3                                      b. 6                                      c. 1                                      d. 9

**Ans.** b. 6

3. How often is the inner loop of a nested loop run?
  - a. Forever
  - b. One fewer time than the main loop
  - c. Each time the main loop is run
  - d. One more time than the main loop

**Ans.** c. Each time the main loop is run

4. Examine the following code. Is there anything that will prevent it from processing?

```
for (int i=1 ; i <= 3; i++) {  
  for (int j=1 ; j <= 3; j++) {  
  }  
}
```

- a. No, it is correct                                      b. No, but it could run forever  
c. Yes, it will return a compiler error                                      d. Yes, but it will show all zeroes in the output

**Ans.** c. Yes, it will return a compiler error

5. What will be the output of the following program?

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

```

{
int sum = 23;
for(int i = 2; i <= 5; i++)
{
for(int j = 7; j <= 9; j++)
{
sum += ( i * j );
}
}
System.out.println("sum= " + sum);
}
}

```

- a. sum= 336
- b. sum= 359
- c. Compilation errors
- d. sum= 45

**Ans.** b. sum= 359

6. What will be the output of the following program?

```

class PrintFormat
{
static void main()
{
for (int i= 0; i < 5; i++)
{
for (int j = 5; j >= i; j--)
{
System.out.print(j);
System.out.println();
}
}
}
}

```

- |          |          |           |          |
|----------|----------|-----------|----------|
| a. 01234 | b. 01234 | c. 543210 | d. 54321 |
| 1234     | 01234    | 54321     | 5432     |
| 234      | 01234    | 5432      | 543      |
| 34       | 01234    | 543       | 54       |
| 4        | 01234    | 54        | 5        |

**Ans.** c. 543210

54321

5432

543

54

7. What will be the output of the following program?

```
class NestedForloop
{
static void main()
{
int sum = 5;
inti= l;
for ( ; i <= 4; )
{
for(int j = 1; j <= 4; j++)
{
sum+= (i * j); i++;
}
}
System.out.println("Sum"+ sum);
}
}
```

a. Sum= 35

b. Sum= 51

c. Sum= 19

d. Compilation Error

**Ans.** a. Sum= 35

8. What will be the output of the following program?

```
class NestedForloop
{
static void main()
{
int sum= 5;
inti= 1;
for ( ; i < 4;)
{
for(int j = 1; j < 4; )
```



```

    {
    for(int k = 1; k < 4; k++)
    {
    sum (i * j);
    i++;
    j++;
    k++;
    }
    }
    System.out.println("Sum =" + sum);
    }
}

```

a. Sum= 35

b. Sum= 576

c. Sum= 16

d. Compilation Error

**Ans. a. Sum= 35**

9. What will be the output of the following program?

```

public class NestedFor
{
static void main()
    {
    for (inti= 0; i < 10; i *= 3)
    {
    System.out.print(i + ":");
    for (i = 6; i < 14; i += 2)
    {
    System.out.print(i + ":");
    }
    }
    }
}

```

a. 0:6:8:10:12:

b. 6:8:10:12:

c. 6:9:12:

d. Some other output or Compilation Error

**Ans. a. 0:6:8:10:12:**

10. Which of the following code segments will produce this output? (Ignore spacing.)

```

2      —      —      —      —      —
—      4      —      —      —      —
—      —      6      —      —      —
—      —      —      8      —      —
—      —      —      —      10     —
—      —      —      —      —      12

```

```

i. for (int i= 1; i <= 6; i++)
{
for (int k = 1; k <= 6; k++)
(k == i)
System.out.print(2 * k);
else
System.out.print("-");
System.out.println();
}

```

```

ii. for (int i= 1; i <= 6; i++)
{
for (int k = 1; k <= i - 1; k++)
System.out.print("-");
System.out.print(2 * i());
for (int k = 1; k <= 6 - i; k++)
System.out.print("-");
System.out.println();
}

```

```

iii. for (int i= 1; i <= 6; i++)
{
for (int k = 1; k <= i - 1; k++)
System.out. print("-");
System.out.print(2 * i);
for (int k = i + 1; k <= 6 - i; k++)
System.out.print("-");
System.out. println();
}

```

- a. i only
- b. i and ii only
- c. ii only
- d. iii only

**Ans.** a. i only

## SECTION A

1. What is a nested loop? Explain with an example.

**Ans.** A loop within another loop is called a nested loop. Program to illustrate the use of nested loop.

```
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.

**2. Illustrate the use of break statement with an example.**

**Ans.** The break statement causes the control flow to exit current **loop** body (as if the **loop** condition has just evaluated to false).

Program to illustrate the use of break statement in for loop.

```
class Loops
{
static void useBreak()
{
    int x;
    for(x=1;x<=10;x++)
    {
        System.out.println(x);
        if (x==5)
            break;
    }
}
```

Output

```
1
2
3
4
5
```

**3. What is continue statement used for? Explain its usage in a nested for loop.**

**Ans.**

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.

In a nested loop the **continue** statement in the inner loop skips the rest of the statement after it and **continue** with the next iteration. In other words the continue statement in a inner-loop will only make the rest of the statements be skipped only of the inner-loop and continue with the next iteration only of the inner loop.

**Program to illustrate the working of continue statement in a nested loop.**

```
public class Loops
{
static void useContinue( )
```

```

    {
    int x,y;
    for(x=1;x<=2;x++)
    {
    for(y=1; y<=4 ; y++)
    {
    if(y==3)
    continue;
    System.out.println(x+"\t"+y);
    }
    System.out.println("x="+x);
    }
    }
}

```

#### 4. How does a nested loop work?

**Ans.**

In a nested loop the inner most loop completes its iteration and only then the control goes to the outer loop. For example,

```

int x , y;
for(x=1;x<=3;x++)
{
    for(y=1; y<=3 ; y++)
    {
        System.out.println(x+ " " +y);
    }
}

```

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.

#### 5. Illustrate nested while and do-while with an example.

**Ans.**

Program to illustrate the use of nested loop using while.

```

public class Loops

```

```

{
static void Nested()
    {
    int x,y;
    x=1;
    while(x<=3)
    {
    y=1;
    while(y<=3)
    {
    System.out.println(x+ " "+y);
    y++;
    }
    x++;
    }
    }
}

```

Program to illustrate the use of nested loop using do-while.

```

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

```

## SECTION B

### A. General Programs involving nested loop.

1. Write a program to input 10 integers and find the sum of the prime numbers only.

**Ans.**

```
import java.util.*;
class Q1
{
static void main()
    {
    int i,j,n,s=0,c;
    Scanner sc=new Scanner(System.in);
    System.out.println("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)
    s=s+n;
    }
    System.out.println("Sum of prime numbers="+s);
    }
}
```

2. Write a program to input 10 integers and find the average of the prime numbers only.

**Ans.**

```
import java.util.*;
class Q2
{
static void main()
    {
    int i,j,n,s=0,c,p=0;
```

```

Scanner sc=new Scanner(System.in);
System.out.println("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)
{
s=s+n;
p++;
}
}
if(p>0)
System.out.println("Average of prime numbers="+(float)s/p);
else
System.out.println("No prime no.s present");
}
}

```

**3. Write a program to input 10 integers and find the sum of the perfect numbers only.**

**Ans.**

```

import java.util.*;
class Q3
{
static void main()
{
int i,j,n,s=0,c;
Scanner sc=new Scanner(System.in);
System.out.println("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+=j;
}
if(c==n)
s=s+n;
}
if(s>0)
System.out.println("Sum of perfect numbers="+s);
else
System.out.println("No perfect no.s present");
}
}

```

**4. Write a program to input 10 integers and find the average of the project numbers only.**

**Ans.**

```

import java.util.*;
class Q4
{
static void main()
{
int i,j,n,s=0,c,p=0;
Scanner sc=new Scanner(System.in);
System.out.println("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+=j;
}
}
}
}

```

```

    if(c==n)
    {
        s=s+n;
        p++;
    }
    }
    if(p>0)
        System.out.println("Average of perfect numbers="+((float)s/p));
    else
        System.out.println("No perfect no.s present");
    }
}

```

**5. Write a program to input 10 integers and find the largest prime number if any.**

**Ans.**

```

import java.util.*;
class Q5
{
    static void main()
    {
        int i,j,n,s=0,c,p=0,l=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("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)
            {
                p++;
                if(p==1)

```

```

l=n;
if (n>l)
l=n;
}
}
if(p>0)
System.out.println("Largest prime numbers="+l);
else
System.out.println("No prime no.s present");
}
}

```

**6. Write a program to input 10 integers and find the smallest perfect number if any.**

**Ans.**

```

import java.util.*;
class Q6
{
static void main()
{
int i,j,n,s=0,c,p=0;
Scanner sc=new Scanner(System.in);
System.out.println("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+=j;
}
if(c==n)
{
p++;
if(p==1)
s=n;
}
}
}
}

```

```

    if (n<s)
    s=n;
    }
    }
    if(p>0)
    System.out.println("Smallest perfect numbers="+s);
    else
    System.out.println("No perfect no.s present");
    }
}

```

**7. Write a program to generate all 2 digit prime numbers.**

**Ans.**

```

class Q7
{
static void main()
    {
    int i,j,c;
    for(i=10;i<=99;i++)
    {
    c=0;
    for(j=1;j<=i;j++)
    {
    if(i%j==0)
    c++;
    }
    if(c==2)
    System.out.println(i);
    }
    }
}

```

**8. Write a program to generate all perfect numbers between 1 to 100.**

**Ans.**

```

class Q8
{
    static void main()

```

```

    {
    int i,j,c;
    for(i=1;i<=100;i++)
    {
    c=0;
    for(j=1;j<i;j++)
    {
    if(i%j==0)
    c+=j;
    }
    if(c==i)
    System.out.println(i);
    }
    }
}

```

9. Write a program to generate all 3 digit palindrome numbers.

**Ans.**

```

class Q9
{
static void main()
{
{
int i,j,r,d;
for(i=100;i<=999;i++)
{
r=0;
for(j=i;j>0;j=j/10)
{
d=j%10;
r=r*10+d;
}
if(r==i)
System.out.println(i);
}
}
}
}

```

10. Input an integer and check whether it is a special number or not. A special number is such a number whose sum of the factorials of each digit is the number itself. For example, 145 is a special number because  $1! + 4! + 5! = 1 + 24 + 120 = 145$ . Special numbers are also called factorion.

Ans.

```
class Q10
{
static void main()
    {
    int i,j,r,d;
    for(i=100;i<=999;i++)
    {
    r=0;
    for(j=i;j>0;j=j/10)
    {
    d=j%10;
    r=r*10+d;
    }
    if(r==i)
    System.out.println(i);
    }
    }
}
```

11. 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 the number, and if this process terminates in 1, then the original number is called a magic number. For example  $55 \Rightarrow (5+5)=10 \Rightarrow (1+0)=1$ .

Ans.

```
import java.util.*;
class Q11
{
static void main()
    {
    Scanner sc=new Scanner(System.in);
    int n,i,s,d;
    System.out.println("Enter a number:");
    n=sc.nextInt();
```

```

for(s=0;n>9;n=s,s=0)
{
for(i=n;i>0;i=i/10)
{
d=i%10;
s+=d;
}
}
if(n==1)
System.out.println("Magic Number");
else
System.out.println("Not a magic number");
}
}

```

- 13. Write a program to input an integer as argument, and print the number by having the digits arranged in ascending order.**

**Ans.**

```

class Q13
{
static void main(int n)
{
int i,s=0,c=0,d,j;
for(i=9;i>=0;i--)
{
for(j=n;j>0;j=j/10)
{
d=j%10;
if(d==i)
s+=d*(int)Math.pow(10,c++);
}
}
System.out.println("New number="+s);
}
}
}

```

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

**Ans.**

```
class Q14
{
    static void main(int n)
    {
        int i,s=0,c=0,d,j,d1,f=0;
        for(i=n;i>0;i=i/10)
        {
            d1=i%10;
            for(j=i/10;j>0;j=j/10)
            {
                d=j%10;
                if(d==d1)
                f=1;
            }
        }
        if(f==0)
            System.out.println("All digits are unique");
        else
            System.out.println("All digits are not unique");
    }
}
```

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

**Ans.**

```
class Q15
{
    static void main(int n)
    {
        int i,c,d,j;
        for(i=0;i<=9;i++)
        {
            c=0;
            for(j=n;j>0;j=j/10)
            {
```



```

    =j%10;
    if(d==i)
        c++;
    }
    if(c>0)
        System.out.println("Frequency of "+i+"="+c);
    }
}

```

### B. Programs involving series using nested loops.

Write programs to find the sum of the following series:

i.  $x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \dots + \frac{x^n}{n!}$

ii.  $\frac{x}{2!} + \frac{x^2}{3!} + \frac{x^3}{4!} + \frac{x^4}{5!} + \frac{x^5}{6!} + \frac{x^6}{7!} + \dots + \frac{x^n}{(n+1)!}$

iii.  $x^2 + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \dots + \frac{x^{n+1}}{n!}$

iv.  $x - \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \dots + \frac{x^n}{n!}$

v.  $\frac{x}{2!} - \frac{x^2}{3!} + \frac{x^3}{4!} - \frac{x^4}{5!} + \frac{x^5}{6!} - \frac{x^6}{7!} + \dots \pm \frac{x^n}{(n+1)!}$

vi.  $x^2 - \frac{x^3}{2!} + \frac{x^4}{3!} - \frac{x^5}{4!} + \frac{x^6}{5!} - \frac{x^7}{6!} + \dots \pm \frac{x^{n+1}}{n!}$

Where x and n are to be taken as input.

**Ans.**

```

i.
import java.util.*;
class QBi
{
static void main()
{
int i,j,x,n,f;

```

```

double s=0;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the value of x:");
x=sc.nextInt();
System.out.println("Enter the value of n:");
n=sc.nextInt();
for(i=1;i<=n;i++)
{
f=1;
for(j=1;j<=i;j++)
f*=j;
s=s+Math.pow(x,i)/f;
}
System.out.println("Sum="+s);
}

```

ii.

```

import java.util.*;
class QBii
{
static void main()
{
int i,j,x,n,f;
double s=0;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the value of x:");
x=sc.nextInt();
System.out.println("Enter the value of n:");
n=sc.nextInt();
for(i=1;i<=n;i++)
{
f=1;
for(j=1;j<=i+1;j++)
f*=j;
s=s+Math.pow(x,i)/f;
}
}

```

```
        System.out.println("Sum="+s);
    }
}
```

iii.

```
import java.util.*;
class QBiii
{
    static void main()
    {
        int i,j,x,n,f;
        double s=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the value of x:");
        x=sc.nextInt();
        System.out.println("Enter the value of n:");
        n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            f=1;
            for(j=1;j<=i;j++)
            f*=j;
            s=s+Math.pow(x,i+1)/f;
        }
        System.out.println("Sum="+s);
    }
}
```

iv.

```
import java.util.*;
class QBiv
{
    static void main()
    {
        int i,j,x,n,f;
        double s=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the value of x:");
```

```

x=sc.nextInt();
System.out.println("Enter the value of n:");
n=sc.nextInt();
for(i=1;i<=n;i++)
{
f=1;
for(j=1;j<=i;j++)
f*=j;
if(i%2==1)
s=s+Math.pow(x,i)/f;
else
s=s-Math.pow(x,i)/f;
}
System.out.println("Sum="+s);
}

```

v.

```

import java.util.*;
class QBv
{
static void main()
{
int i,j,x,n,f;
double s=0;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the value of x:");
x=sc.nextInt();
System.out.println("Enter the value of n:");
n=sc.nextInt();
for(i=1;i<=n;i++)
{
f=1;
for(j=1;j<=i+1;j++)
f*=j;

```

```

        if(i%2==1)
            s=s+Math.pow(x,i)/f;
        else
            s=s-Math.pow(x,i)/f;
        }
        System.out.println("Sum="+s);
    }
}

```

vi

```

import java.util.*;
class QBvi
{
    static void main()
    {
        int i,j,x,n,f;
        double s=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the value of x:");
        x=sc.nextInt();
        System.out.println("Enter the value of n:");
        n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            f=1;
            for(j=1;j<=i;j++)
                f*=j;
            if(i%2==1)
                s=s+Math.pow(x,i+1)/f;
            else
                s=s-Math.pow(x,i+1)/f;
        }
        System.out.println("Sum="+s);
    }
}

```

### C. Patterns

Using nested loops write programs for each of the following patterns on the screen:

- |            |             |        |           |           |       |
|------------|-------------|--------|-----------|-----------|-------|
| i. 1       | ii. 5       | iii. 1 | iv. 54321 | v. 54321  | vi. 1 |
| 12         | 54          | 21     | 5432      | 4321      | 22    |
| 123        | 543         | 321    | 543       | 321       | 333   |
| 1234       | 5432        | 4321   | 54        | 21        | 4444  |
| 12345      | 54321       | 54321  | 5         | 1         | 55555 |
| vii. 55555 | viii. 11111 | ix. 5  | x. *      | xi. ***** |       |
| 4444       | 2222        | 44     | **        | ****      |       |
| 333        | 333         | 333    | ***       | ***       |       |
| 22         | 44          | 2222   | ****      | **        |       |
| 1          | 5           | 11111  | *****     | *         |       |
- 
- |        |             |  |  |  |
|--------|-------------|--|--|--|
| xii. 1 | xiii. 12345 |  |  |  |
| 21     | 2345        |  |  |  |
| 321    | 345         |  |  |  |
| 4321   | 45          |  |  |  |
| 54321  | 5           |  |  |  |
- 
- |        |       |        |         |
|--------|-------|--------|---------|
| xiv. * | xv. * | xvi. * | xvii. * |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
| * *    | * *   | * *    | * *     |
- 
- |              |        |       |            |             |
|--------------|--------|-------|------------|-------------|
| xviii. 10101 | xix. 1 | xx. 1 | xxi. 12345 | xxii. 12345 |
| 01010        | 01     | 11    | 23451      | 12341       |
| 10101        | 101    | 101   | 34512      | 12312       |
| 01010        | 0101   | 1001  | 45123      | 12123       |
| 10101        | 10101  | 10001 | 51234      | 11234       |
- 
- |              |             |            |
|--------------|-------------|------------|
| xxiii. 12345 | xxiv. 54321 | xxv. 12345 |
| 12341        | 15432       | 11234      |
| 12322        | 21543       | 22123      |
| 12333        | 32154       | 33312      |
| 14444        | 43215       | 44441      |

Ans. i.

```

class QCi
{
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 QCii  
{  
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 QCiii  
{  
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 QCiv
{
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 QCv
{
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 QCvi
{
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 QCvii
{
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 QCviii
{
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 QCix
{
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 QCx
{
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 QCxi  
{  
    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 QCxii  
{  
    static void main()  
    {  
        int i,j;  
        for(i=1;i<=5;i++)  
        {  
            for(j=1;j<=5;j++)  
            {  
                if(i+j>=6)  
                    System.out.print(6-j);  
                else  
                    System.out.print(" ");  
            }  
        }  
    }  
}
```

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

xiii.

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

xiv.

```
class QCxiv
{
    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();
}
}
}
}
}
}
}
}
}
}

```

xv.

```

class QCxv
{
    static void main()
    {
        int i,j;
        for(i=1;i<=5;i++)
        {
            for(j=1;j<=5;j++)
            {
                if(i==3 || j==3)
                System.out.print("*");
                else
                System.out.print(" ");
            }
            System.out.println();
        }
    }
}
}

```

xvi.

```

class QCxvi
{
    static void main()
    {
        int i,j;
        for(i=1;i<=5;i++)
        {
            for(j=1;j<=9;j++)
            {
                if(i==j || i+j==10)

```

```

        System.out.print("*");
    else
        System.out.print(" ");
    }
    System.out.println();
}
}
}

```

xvii

```

class QCxvii
{
static void main()
    {
    int i,j;
    for(i=1;i<=5;i++)
    {
    for(j=1;j<=9;j++)
    {
    if(i+j==6 || j-i==4)
    System.out.print("*");
    else
    System.out.print(" ");
    }
    System.out.println();
    }
    }
}

```

xviii.

```

class QCxviii
{
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();
    }
}

```

xix.

```

class QCxix
{
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();
    }
    }
}
}

```

xx.

```

class QCxx
{
static void main()
{
    {
    int i,j;
    for(i=1;i<=5;i++)

```

```

    {
    for(j=1;j<=i;j++)
    {
    if(j==1 || i==j)
    System.out.print(1);
    else
    System.out.print(0);
    }
    System.out.println();
    }
}

```

xxi.

```

class QCxxi
{
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 QCxxii
{
static void main()
{
int i,j;
for(i=1;i<=5;i++)
{

```



```

    for(j=1;j<=6-i;j++)
    System.out.print(j);
    for(j=1;j<=i-1;j++)
    System.out.print(j);
    System.out.println();
    }
    }
}

```

xxiii.

```

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

```

xxiv

```

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

```

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

xxv.

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

# Chapter 11

## Computing and Ethics



### A Tick (✓) the correct option.

1. The term Computer Ethics is also referred as:

- a. Cyberethics
- b. Information ethics
- c. Global information ethics
- d. All of these

**Ans.** d. All of these

2. Name the categories used to specify Intellectual Property Rights.

- a. Copyright
- b. Industrial Property
- c. Both a and b
- d. None of these

**Ans.** c. Both a and b

3. Right to privacy is part of which article of the Indian Constitution?

- a. Article 12
- b. Article 19
- c. Article 21
- d. Article 31

**Ans.** c. Article 21

4. What is the term used to refer to sending unwanted bulk messages, especially advertising, indiscriminately?

- a. Phising
- b. Spamming
- c. Virus
- d. Malware

**Ans.** b. Spamming

5. What is the right given to freedom from intrusion by the public, especially as a right called?

- a. Right to Privacy
- b. Copyright
- c. Intellectual Property Rights
- d. None of these

**Ans.** a. Right to Privacy

6. Name the term that refers to a technique used to gain unauthorized access to a computer system or network and in some cases even making unauthorized use of this access.

- a. Virus
- b. Antivirus
- c. Malware
- d. Hacking

**Ans.** d. Hacking

7. Name the type of Malware that prevents the user from using the system, and may destroy files that were open at the time of the attack.
- a. Denial of Service (DoS) Attacks
  - b. Access Violations
  - c. Both a and b
  - d. None of these

**Ans.** a. Denial of Service (DoS) Attacks

8. Where does Malicious Code Hide?
- a. Email
  - b. Web Content
  - c. Legitimate Sites
  - d. All of these

**Ans.** d. All of these

9. Name the program that contains hidden malicious functions but look like something funny or useful such as a game or utility, but harms the system when executed.
- a. Trojans
  - b. Virus
  - c. Spyware
  - d. All of these

**Ans.** a. Trojans

10. Name the program that secretly gather information from the computer it has infected and transmit the information to a different location.
- a. Trojans
  - b. Virus
  - c. Spyware
  - d. All of these

**Ans.** c. Spyware

## **B. Fill in the blanks.**

1. Thus **Intellectual property** rights may be defined as the rights given to persons over the creations of their minds.
2. Right to **privacy** refers to the freedom from intrusion by the public, especially as a right.
3. **Spamming** refers to the use of electronic messaging systems to send unwanted bulk messages, especially advertising, indiscriminately.
4. The term **Piracy** refer to unauthorized copying of software.
5. **Cybercrime** is a term which is used to refer to criminal activity performed using computers and the Internet.
6. **Worms** are programs that do not depend upon another program to replicate it but takes the power of networking to spread itself and also executes independently.
7. **Adware** comes bundled with some commercial software or freeware, which when installed, installs packages that download advertising materials and display them.
8. **Hijack** This is a malicious program that hijacks your browser.
9. **Spyware** programs secretly gather information from the computer it has infected and transmit the information to a different location.
10. *Commercial programs that are made available to the public illegally are often called **warez**.*

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

1. Buying legal software from reputed retailers or organizations is a way to stop Software Piracy. T
2. Spamming refers to the use of electronic messaging systems to send unwanted bulk messages especially advertising, indiscriminately. T
3. Copyright refers to the freedom from intrusion by the public, especially as a right. F
4. Right to Privacy are given to authors of literary and artistic works such as books and other writings, musical compositions, paintings, sculpture, computer programs and films. F
5. Copyright is a one area of Intellectual Property Rights. T
6. Quarantine is used when certain virus may be too difficult to be removed completely, this virus are disabled and kept in a safe place so that it cannot infect other files. T
7. Spyware is a software that is used to remove viruses and other malware programs from your computer. F
8. Push technology enables news and other content providers to automatically supply subscribers with information by downloading content to the user's desktop. T
9. Denial of Service (DoS) attacks steal away login names and passwords, credit card/debit card numbers, files, or other confidential data. F
10. Creating and distributing viruses and/or malicious code on other computers or posting confidential business information on the Internet are also part of cybercrime. T

**SECTION A**

**Answer the following Questions.**

**1. What is Computer Ethics?**

**Ans.** Computer ethics deals with the procedures, values and practices that govern the process of consuming computing technology and its related disciplines without damaging or violating the moral values and beliefs of any individual, organization or entity.

**2. What is Software Piracy? How does shareware deal with software piracy?**

**Ans.** Software piracy is used to describe the act of illegally using, copying or distributing software without ownership or legal rights.

Shareware is a good way to market software. It allows consumers to evaluate an application prior to making a purchase decision. They can easily determine if it meets their business or personal needs, which usually results to a satisfied customer.

**3. Why do you think in developing countries like India, it is difficult to stop software piracy?**

**Ans.** Software piracy is hard to stop for several reasons:

- a. It's not a violent crime, so eliminating the criminals by killing them is not really an option. As such, the only remaining option is a legal approach - charging them with a crime and threatening them with prison if they don't quickly agree to a guilty plea.

- b. Software piracy is very easy. If you want your software to work without internet connection, then you need to put the entire code and all the data into the hands of a users; consequently, any anti-piracy measures have to be in their hands as well, making it possible for smart coders to reverse-engineer your code, find the part which performs verification of “license to use”, and remove or twist that part so that the software works without a license.
- c. Sharing cracked software is ludicrously easy and hard to detect.
- d. There are millions of people involved in this. Quite literally, not only your prisons, but your court-rooms as well are not big enough to realistically charge everyone. So you have to prioritize who you will go after.
- e. Experts still can’t agree to what extent is piracy harmful. There were numerous cases where a pirated version of something was instrumental in making it popular, which not only later led people to buy a legal copy, but generated interest in a sequel, which then became massive hit.
- f. People who can afford buying movies and games generally do so; people who cannot afford it will either pirate them, or not get them at all.
- g. The price tag associated with a legal software is generally high.

**4. What are the different ways of stopping Software Piracy?**

**Ans.** The different ways to stop software piracy are:

- Educate your staff on the licensing requirements of your software purchases
- Conduct a self-audit of your software licenses
- Acquire any licenses needed for full compliance
- The most widely used method is the license key; code that is built into an application to require a valid key to unlock the software.

**5. How does Spamming affect economically?**

**Ans.** Spamming remains economically viable because advertisers have very little or sometimes no operating costs beyond the management of their mailing lists, and it is almost impossible to hold senders accountable for their mass mailings. On the other hand it costs huge to the sender and may sometime even get dubbed.

**6. Discuss two main areas of Industrial Property.**

**Ans.** Copyright and Trademark.

**7. How can spamming be reduced?**

**Ans.** By using these precautions you can greatly mitigate what spam you do receive and prevent most spam from ever happening.

- Be careful where you enter your email at.
- Create or use disposable email addresses for websites you do not trust.
- Never open spam when you receive it.
- Keep your computer virus and malware free.

- If your friends are sending you emails sent to a large recipient list, request that they use BCC instead of TO or CC, so that other recipients cannot see your email address; or request they stop including you if you do not want to receive the emails.
- Do not list your email address on your website or anywhere the public can access it.

**8. How is phishing and pharming performed to perform Cybercrime?**

**Ans.** Phishing is a fraudulent practice of sending emails purporting to be from reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers.

Pharming is the fraudulent practice of directing Internet users to a bogus website that mimics the appearance of a legitimate one, in order to obtain personal information such as passwords, account numbers, etc.

**9. What are the different types of Cybercrimes?**

**Ans.** The different types of cyber crime are:

- Financial fraud crimes
- Cyberterrorism
- Cyberextortion
- Cyberwarfare
- Computer as a target
- Computer as a tool

**10. How are Hackers different from Crackers?**

**Ans.** Hackers are those computers experts which breaks into computers to check any vulnerably so that no one can misuse the services. These are really intelligent and smart persons who use there ability to protect the community from cyber crimes and computer thefts.

Whereas, Crackers are those peoples who use there knowledge to do computer crimes for gaining popularity among peoples and to earn fast money. They break into computer networks for their enjoyment and cause harm to them. These persons does not have real knowledge and know something about using the particular software to break into computers.