Class IX

Answer Key

Computer Applications

1. Principles of Object-Oriented Programming

Unsolved Questions

A. 1. c 2. a 3. c 4. b 5. a

B. 1. Global Variables 2. Compiler
3. Procedure Programming 4. Programming Paradigm
5. Object-Oriented Language

C. 1.

	POP (Procedure-Oriented Programming)	OOP (Object-Oriented Programming)
Complexity Management	Becomes harder to manage as the program grows in size.	Handles complexity better using modularity and abstraction.
Core Concept	Focuses on procedures (functions) to perform operations.	Focuses on objects that encapsulate data and behavior.
Data Handling	Data is exposed and shared globally, making it less secure.	Data is encapsulated within objects, ensuring security.
Modularity	Divides programs into functions.	Divides programs into objects and classes.
Real-World Modeling	Less natural for real-world problem modeling.	Mimics real-world entities and interactions.
Reusability	Limited reusability; functions must be adapted for reuse.	Promotes reusability through inheritance and polymorphism.
State and Behavior	Functions act on global data.	State (attributes) and behavior (methods) are bound together in objects.
Structure	Procedural, with functions called sequentially or hierarchically.	Organized around objects interacting with each other.
Examples of Languages	C, Pascal, Fortran.	C++, Java, Python, C#.

2. In Object oriented programming, **polymorphism** is the feature of being able to allot a dissimilar meaning so that a variable, a method or an object can have more than one form.

   Real-Life Example:

   A Person: A single person can play different roles in different contexts:

   - At home, they act as a parent (nurturing, guiding).
   - At work, they act as an employee (performing specific tasks).
   - With friends, they act as a friend (socializing, having fun).

   The person is the same, but their behavior changes based on the situation. This is analogous to polymorphism, where an object or method can take on multiple forms.

3. Disadvantages of Object-Oriented Programming
   1. Object oriented program needs more lines of coding than procedural oriented program.
   2. It needs more memory to execute at a high speed. Thus, its execution is slower than the conventional encoding system.
   3. It requires intensive testing.
   4. Proper planning is necessary for Object oriented programming.

4. Difference between polymorphism and encapsulation

	Polymorphism	Encapsulation
Definition	Allows an entity (method, object, or operator) to behave differently based on the context.	Bundles data and methods into a single unit and restricts access to certain components.
Purpose	Achieves flexibility and reusability by defining common interfaces or behaviors.	Ensures data security by restricting direct access and hiding implementation details.
Focus	Focuses on behavior (methods or functions) and how it can adapt to different situations.	Focuses on data (attributes) and controlling how it is accessed or modified.
Implementation	Achieved through method overloading, method overriding, or operator overloading.	Achieved using access modifiers like private, protected, and public.
Code Reusability	Promotes code reusability by allowing common behaviors across different classes.	Makes code maintainable and secure by restricting direct access to data.

5. Real-World Example of inheritance:

   Banking System:

   - Parent class: Account (common attributes: balance, account number).
   - Child classes: SavingsAccount, CurrentAccount, each with unique behaviors like interest calculation or overdraft.

6. Difference between Machine Language and Assemble Language:

	Assembly Language	Machine Language
Definition	A low-level language that uses human-readable mnemonics to represent machine instructions.	The lowest-level programming language, consisting of binary (0s and 1s) instructions understood directly by the CPU.
Readability	Easier to read and understand due to the use of mnemonics (e.g., ADD, MOV).	Difficult to read and understand as it consists of binary or hexadecimal codes.
Representation	Symbolic, uses mnemonics and labels (e.g., LOAD A).	Pure binary (e.g., 10001101) or hexadecimal (e.g., 8D).
Ease of Use	Easier to write, debug, and maintain compared to machine language.	Harder to write and prone to errors because of its binary representation.

7. 2 Advantages of Polymorphism
   a. Code Reusability: Polymorphism allows you to write one function or method that works for different types of objects. This reduces code duplication and makes your program cleaner and easier to maintain.
   b. Flexibility and Easy Extensibility: Polymorphism makes it easy to add new features or objects to your program without modifying existing code. This makes your program flexible and future-proof.
8. 2 Disadvantages of Machine Leven Language
   - Hard to Write and Understand: Machine language is written in binary (0s and 1s), which is very difficult for humans to read, write, or debug.
   - Not Portable: Machine language works only on the specific type of computer or processor it's written for. If you change the hardware, you need to rewrite the program.

D. 1. b 2. d 3. a 4. a 5. b

E. 1. b 2. a 3. a 4. b 5. a

F. 1. c 2. a 3. c

2. Introduction to Java

Unsolved Questions

A. 1. d 2. d 3. b 4. b 5. a

B. 1. machine code 2. object 3. Bluej
4. independent 5. automatic backup

C. 1. BlueJ is an integrated development environment (IDE) specifically designed for teaching and learning Java programming. It is widely used in educational institutions to help beginners understand object-oriented programming concepts.

2. Java Applications: It is a java program that runs independently on a computer without any external help or internet connections.

   Applets are Java programs which are downloaded from a website on a client computer and can only be executed within a web browser. They cannot run independently on a computer like standalone applications.

3. To create a new project, follow the given steps:

   Step 1: Click Project menu from Menu bar.

   Step 2: Select the New Project option from the drop-down menu. The New Project dialog box opens.

   Step 3: Type a name for your project in the Name text box.

   Step 4: Choose a location by clicking the choose button.

   Step 5: Click the OK button. A project is created in the BlueJ application window

4. Disadvantages of Java:

   a. Time-consuming: Java is both compiled and interpreted. Since java runs on web, the compiled program send from the server and has to be interpreted in client machine by JVM to convert to machine-depende language. This makes Java significantly slower than C or C++. Also garbage collector consumes more CPU ti and leads to poor performance.

   b. Unattractive Look: Though in Java there are many GUI builders to create a graphical interface, but they are appropriate and many discrepancies are visible when used.

   c. Lack of Backup Facility: Java does not provide any options of backing up of data which is one of the ma reasons for low ratings among users.

   d. Required Large Memory Space: Requirement of memory space is much higher as compared to other language like C and C++.

5. Java was developed by James Gosling and his team including Mike Sheridan and Patrick Naugton. It was initially known as OAK.

D. 1. a 2. d 3. d 4. a 5. a

E. 1. C++ Language 2. James Gosling

3. Sun Microsystems 4. Write Once, Run Anywhere

5. Developing Apps, Creating websites contents, games and softwares.

F. 1. d 2. c 3. d

3. Elementary Concept of Objects and Classes

Unsolved Questions

A. 1. a 2. a 3. a 4. b 5. a 6. c

B. 1. Method invocation 2. Object 3. Class

4. Instant variable 5. Blue print/ template

6. a. sub_name b. instance variable c. getData(), showData()

d. subject e. computer f. new

C. 1. Tata nano, Alto 800

2. Data Member and Method

3. In Object-Oriented Programming (OOP), a class is a blueprint or template for creating objects. It defines the attributes (data members) and behaviors (methods) that objects of that class will have.

4. An object is an instance of a class because it is a concrete realization of the blueprint defined by the class. When an object is created, it has its own unique values for the attributes defined in the class and can use the class's methods.

5. Message passing occurs when an object calls the method of another object, which can lead to the exchange of data or triggering specific behaviors this mechanism is fundamental in object oriented programming as it allows object to collaborate and perform complex tasks.

6. An object is an instance of a class that has a unique identity, state (attributes), and behavior (methods). It is a real-world entity that interacts with other objects in an object-oriented program.

   Different types of characteristics of an Object:

   1. State – Defined by attributes (variables).
   2. Behavior – Defined by methods (functions).
   3. Identity – A unique reference in memory.

   Example:

```
class Car {
    String brand;
    int speed;

    // Constructor
    Car(String b, int s) {
        brand = b;
```

```
        speed = s;
    }

    // Method to display car details
    void display() {
        System.out.println ("Brand: " + brand + ", Speed: " +
        speed + " km/h");
    }
}
```

Example

```
Public class Object {
    public static void main(String[] args) {
        Car car1 = new Car("Toyota", 120); // Creating an object
        car1.display();  // Calling the method
    }
}
```

7. A class is called a factory of objects because it acts as a blueprint or template from which multiple objects can be created. Just like a factory produces multiple products of the same type, a class allows the creation of multiple objects with similar properties and behaviors.
8. A class is a user-defined data type because it allows programmers to create new types beyond built-in types. These new types (classes) can hold multiple variables and methods, making code more modular and reusable.

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

E. 1. (i) b (ii) d

2. (i) b (ii) b (iii) b

F. b

4. Values and Types

Unsolved Questions

A. 1. a 2. b 3. c 4. c 5. a 6. b

7. d 8. c 9. d 10. c 11. a 12. b

13. a 14. c 15. a 16. b

B.

1. Different	2. User defined	3. Smaller
4. null	5. -128 to 127	6. char ch='A';
7. static	8. fixed	

C. 1. Literals in Java are values that are assigned to a variable. They are used by programmers and can be of any data type. The other name of Literals is "constants".

a. Real Literal: Real literals (also known as floating-point literals) are used to represent decimal (fractional) values in Java. They belong to the float and double data types.

b. Boolean Literal: A Boolean literal in Java represents one of the two possible truth values. They are true and false. These literals belong to the boolean data type and are used for logical conditions.

2.

Declaration	Initialization
• Declaring a variable means specifying its type and name.	• Assigning a value to a declared variable.
• Memory is reserved, but no value is assigned.	• The variable gets an actual value.
• dataType variableName;	• variableName = value;
• int num;	• num = 10;
• Cannot be used before initialization (will cause an error if accessed).	• The variable is ready for use.

3. Use of

a. `\t`: It is an escape sequence. It is called Horizontal Tab and used to separate two values on the screen as output with a gap of tab spaces between them. It is used mainly to print according to a given format.

Eg. System.out.println("Good Moring:\t Students");

Output: Good Moring: Students

b. `\n`: The \n escape sequence is used to insert a new line (line break) in Java. It moves the cursor to the next line in the output.

Eg. System.out.println("Hello\nWorld!");

Output: Hello
World!

4. Operators in Java are symbols that perform arithmetical or logical calculations on one or more operands. They are classified into different types based on their functionality. There are three types of operators in Java:

a. Arithmetic Operators

b. Relational Operators

c. Logical Operators

5. a. Punctuators are special characters in java used as punctuation signs. Some examples are
   - ?
   - :
   - ;
   - .

   b. Separators are the special characters in Java used to separate the variables or characters in a Java Programs. Some examples:
   - ,
   - ()
   - {}
   - []
6. The size of the following terms of bits:
   a. short: 16 bits
   b. double: 64 bits
   c. int: 32 bits
   d. char: 16 bits
7. Escape sequence are non-graphical character that is preceded by a backslash(\) and has a special meaning for the compiler. There are eight escape sequences in Java each performing special task.

   Example: `\t` – Horizontal Tab and `\n` New Line feed.
8. a. Explicit Type Casting

   b. Implicit Type Casting
9.

Primitive Data Types	Non-Primitive Data Types
• Basic built-in data types	• User-defined or Java-provided objects
• int, char, float, boolean	• String, Array, Class, Interface
• Fixed (depends on type)	• Variable (depends on object)
• Defaults to 0, false, or null (for char)	• Defaults to null

10. A composite data type (also called a complex or non-primitive data type) is a type that can hold multiple values or a collection of data. These types are built using primitive data types and provide structured storage.

    Example of Composite Data Type : Array, Class etc.

D. 1. a 2. a. 3. a 4. a 5. a

E. 1. b 2. d

5. Operators in Java

Unsolved Questions

A.

1. a	2. c	3. b	4. a	5. c	6. b
7. c	8. c	9. a	10. b	11. c	12. d
13. a	14. b	15. b			

B. 1. Operator 2. .(dot) 3. new 4. Boolean 5. one, varyiny

C. 1. m = 15, n = 5, r = 0

r = 15/6

= 2

r = ++m + ++n – n++

= 16+6-6

= 16

2. Arithmetic Operators: An arithmetic operator is a symbol used in mathematics and programming to perform basic arithmetic operations on numerical values. The common arithmetic operators include Addition (+),Subtraction (-),Multiplication (*),Division (/) etc.

3. Java Expression:
   - A=(B+C)/2.0*h
   - V=3.142*r*r*h

4. i. a = ++a + a-- / ++a + --a

= 3+3/3+2

= 3+1+2

= 6

ii. i* = j++%j--/k*10

I = i*(j++%j--/k*10)

= 2*(4%5/3*10)

= 2*(4/3*10)

= 2*(1*10)

= 2*10

= 20

5. a. Relational Operator

   b. Increment operator

c. Logical Operator

d. Ternary Operator

6. 75 F

7. A counter and an accumulator are both used in programming to store and update values, but they are used for different purposes.
   - Counter: A counter is used to keep track of occurrences, such as counting loops, events, or objects. It typically increases (or decreases) by 1.
   - Accumulator: An accumulator is used to keep a running total or cumulative sum of values. It usually increases by a variable amount, depending on the data being processed.

8. Difference between:

   a. **Arithmetic operators** are used to perform mathematical calculations such as addition, subtraction, multiplication, and division. Arithmetic operators work with numerical values and produce numerical results. Arithmetic operators are used in calculations, such as computing totals, averages, and financial figures. and Arithmetic operations return numeric values.

   Example 5 + 3 = 8 (Addition), 10 / 2 = 5 (Division).

   Logical operators are used to evaluate logical expressions and return True or False based on conditions. Logical operators work with Boolean values (True or False) and determine logical relationships between expressions. Logical operators are used in decision-making, conditional statements, and controlling program flow (e.g., if statements in programming). Logical operations return Boolean values (True or False).

   Examples (5 > 3) and (2 < 4) $\rightarrow$ True, not (5 > 10) $\rightarrow$ True

   b. **Logical AND (and)** requires all conditions to be true for the final result to be True. AND checks if all conditions are met. If even one condition is False, the result is False.

   Logical OR (or) requires at least one condition to be true for the final result to be True. OR checks if at least one condition is met. If at least one condition is True, the result is True.

9. The prefix increment operator is a type of increment operator used in programming languages like Java. It increases the value of a variable by 1 before using it in an expression.

10. m = 6

    n = 14

D. 1. b 2. b 3. a 4. c 5. b

E. 1. d 2. a 3. b 4. c 5. b

F. 1. d 2. a

6. Input in Java

Unsolved Questions

A. 1. a 2. a 3. a 4. d 5. c

B. 1. Syntax, Logical, Runtime 2. Execution 3. String 4. float
5. debugging

C. 1. Difference between Single-line comment, Multiline comment and Document comment:

- **Single-Line Comment:** Used for short explanations, placed on a single line.

  Eg. `// This is a single-line comment`

- **Multi-Line Comment:** Used for longer comments spanning multiple lines, often for temporary notes or to disable code.

  Eg. `/* This is a multi-line comment */`

- **Document Comment:** A special comment used to generate documentation for classes, methods, and fields.

  Eg. `/** This is a Javadoc comment */`

2. Difference between Syntax error, Logical error and Runtime error:

- **Syntax Error (Compilation Error):** It occurs when Java code violates language rules such as syntax mistakes like missing semicolons, incorrect data types, and invalid keywords. It prevents the program from compiling.

  For example: `int num = "hello"; // Error: Type mismatch`

- **Logical Error:** The program runs but produces incorrect results due to flawed logic. These errors are the hardest to detect.

  For example:
  ```
  int result = 10 / 2 * 2; // Intended: (10 / (2 * 2)) =
          2, but result = 10
  ```

- **Runtime Error:** It occurs while the program is running, usually due to invalid operations like division by zero or accessing an invalid index.

  For example: `int num = 5 / 0; // Arithmetic`

  Exception: `/ by zero`

3. InputStreamReader is used to input values from the users whenever required at the time of execution. To use this class, we need another class called BufferedReader class that aloows us to read input line by line. For this class we need to include a package called "java.io".

4. Difference between try and catch:

- **Try block:** It contains the code that might throw an exception. When an exception occurs within the try block, the normal flow of execution is interrupted, and the runtime looks for an appropriate catch block to handle the exception.

- **Catch Block:** It handles the exception thrown from the try block. When an exception is caught, the catch block is executed, allowing you to manage the error (log it, display a message, recover, etc.) without crashing the program.

5. Java Comments: In Java, comments are annotations in the source code that are ignored by the compiler and runtime environment. They serve to document the code, making it more understandable for developers by providing explanations, clarifications, or notes.
6. Example of Runtime error:

```
int a = 10;
int b = 0;
int result = a / b; // Throws Arithmetic
Exception: / by zero
```

7. The methods of Scanner class are:
   a. `nextInt()`
   b. `nextLine()`
   c. `next().charAt(0)`
   d. `nextDouble()`
   e. `nextShort()`
8. Debugging is the process of fixing the errors found during testing. It focuses on correcting these issues to ensure the code works as expected.

D. a

E. 1. import java.util.*;

```
class prog1
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int a, b,t;
        System.out.print("Enter 1st number : ");
        a=sc.nextInt();
        System.out.print("Enter 2nd number : ");
        b=sc.nextInt();
        System.out.println("Before interchanging : ");
        System.out.println("a = "+a + " : b= "+b);
        t=a;
        a=b;
        b=t;
```

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

    // Output
    Enter 1st number : 10
    Enter 2nd number : 20
    Before interchanging :
    a = 10 : b= 20
    a = 20 : b= 10
```

2. class prog2

```
    {
        public static void main(double cp)
        {
            double p,sp,pp;
            p=50;
            sp=cp+p;
            pp=p/cp*100;
            System.out.println("Selling Price : "+ sp);
            System.out.println("Profit Percentage : "+pp);
        }
    }
    //Output
    Selling Price : 90.0
    Profit Percentage : 125.0
```

3. import java.util.*;

```
    class prog3
     {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            // Prompt the user to enter the number of days
            System.out.print("Enter the number of days: ");
            int t_Days = sc.nextInt();
            // Constants for conversion
            final int DIY = 365;
            final int DIM = 30;
```

```
        // Calculate the number of years
        int years = t_Days / DIY;

        // Calculate the remaining days after extracting years
        int r_Days_Years = t_Days % DIY;

        // Calculate the number of months
        int months = r_Days_Years / DIM;

        // Calculate the remaining days after extracting months
        int r_Days = r_Days_Years % DIM;

        // Display the result
        System.out.println(t_Days + " days is approximately " +
        years + " years, " + months + " months, and " + r_Days + "
        days.");
    }
}
//Output
Enter the number of days: 260
260 days is approximately 0 years, 8 months, and 20 days.
```

4. class prog4

```
{
    public static void main(double s)
    {
        double d,p;
        d=Math.sqrt(2)*s;
        p=4*s;
        System.out.println("Diagonal : "+d);
        System.out.println("Perimeter : "+p);
    }
}
//Output if side is 10
Diagonal : 14.1421
Perimeter : 40.0
```

5. import java.util.*;

```
class prog5
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String name;
        double dailyWage, m_inc,tax=0.0;
        // Input: Person's name
        System.out.print("Enter your name: ");
        name = sc.nextLine();
        // Input: Daily wage
        System.out.print("Enter your daily wage in Rs.: ");
        dailyWage = sc.nextDouble();
        // Calculate gross monthly income
        m_inc = dailyWage * 30;
        if (m_inc > 10000)
        {
            tax = 500;
            m_inc = m_inc-500;
        }

        // Display the results
        System.out.println("Income Details for " + name + ":");
        System.out.println("Tax Deducted: Rs. " + tax);
        System.out.println("Gross Monthly Income: Rs. " + m_inc);
    }
}
//Output
Enter your name: Amit
Enter your daily wage in Rs.: 1020
Income Details for Amit:
Tax Deducted: Rs. 500.0
Gross Monthly Income: Rs. 30100.0
```

6. import java.util.*;

```
class prog6
{
```

```
    public static void main()
    {
        Scanner sc=new Scanner(System.in);
        String name;
        double pr,dis,pr_dis,gst,total_amt;
        System.out.println("Enter product details");
        System.out.print("Enter name : ");
        name=sc.next();
        System.out.print("Enter price : ");
        pr=sc.nextDouble();
        dis=0.15*pr;
        pr_dis=pr-dis;
        gst=0.18*pr_dis;
        total_amt=pr_dis+gst;
        System.out.println("\t Bill Amount");
        System.out.println("Name "+"\t"+"Price");
        System.out.println(name+"\t"+total_amt);
    }

}

//Output
Enter product details
Enter name : Amit
Enter price : 10000
  Bill Amount
Name    Price
Amit    10030.0
```

7. import java.util.*;

```
class prog7
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        double dis_m, dis_km, total_cost;
        System.out.print("Enter the distance in m : ");
        dis_m=sc.nextDouble();
```

```
            dis_km=dis_m/1000.0;
            total_cost=dis_km*10;
            System.out.println("Total Distance in Km : " + dis_km);
            System.out.println("Total Amount : "+total_cost);
        }
}

//Output
Enter the distance in m : 1234
Total Distance in Km : 1.234
Total Amount : 12.34
```

8. import java.util.*;

```
class prog8
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        double f,c;
        int ch;
        System.out.println("Enter 1 for Farenheit to Celcius :");
       System.out.println("Enter 2 for Centigrade to Farenheit :");
        ch=sc.nextInt();
        if(ch==1)
        {
            System.out.print("Enter temperature in Farenheit : ");
            f=sc.nextDouble();
            c=(f-32)/9.0*5;
            System.out.print("Temperature in Celcius : "+c);
        }
        if(ch==2)
        {
            System.out.print("Enter temperature in Celcius : ");
            c=sc.nextDouble();
            f=c/5.0*9.0+32;
            System.out.print("Temperature in Farenheit : "+f);
        }
    }
```

```
}
//Output
Enter 1 for Farenheit to Celcius :
Enter 2 for Centigrade to Farenheit :
1
Enter temperature in Farenheit : 66
Temperature in Celcius : 18.88888888888889
```

7. Mathematical Library Methods

Unsolved Questions

A. 1. b 2. b 3. a 4. c 5. c 6. c
7. b 8. c 9. a 10. d

B. 1. 5.0 2. Math.cbrt() 3. double 4. 10.0
5. Math.sqrt(Math.pow(a,2)+Math.pow(b,3)) 6. 2*a*b*c*Math.sqrt(a*b))
7. 3.0 8. 2 9. -5.5 10. double

C. 1. The difference between library method and user-defined method are:

- **Library Method :** Pre-built functions readily available in programming language libraries. They are used for common and frequently used operations. for example : Math.sqrt() in Java (calculates the square root of a number)
- **User-defined method :** Functions created by the programmer to meet specific program requirements. Tailored to address unique tasks or functionalities within a program.

  A method to calculate the area of a custom shape.

```
void area(int l, int b)
{
int a=l*b;
System.out.println("Area "+a);
}
```

2. Example of Math.max() : Math.max(4,5) : Output : 5.0
   Example of Math.max() : Math.max(4.5,6.7) : Output : 6.7
   Example of Math.pow() : Math.pow(3,2) : Output : 9.0
3. Only 1 argument
4. A = P*(Math.pow((1+r/n),t);

5. The difference between Math.ceil() and Math.floor()

   Math.ceil(double x) : Rounds up to the nearest integer and Returns a double value.

   Example:

   1. System.out.println(Math.ceil(4.2)); // Output: 5.0

      System.out.println(Math.ceil(-4.8)); // Output: -4.0

   2. Math.floor(double x) : Rounds down to the nearest integer and Returns a double value.

      System.out.println(Math.floor(4.9)); // Output: 4.0

      System.out.println(Math.floor(-4.2)); // Output: -5.0

6. 3.0
7. -1.0

D. a

E. 1. import java.util.*;

```
class chapter7_1
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        double pi=22.0/7.0,r,sl_ht,ht,vol,sur_ar;
        System.out.print("Enter radius of cone : ");
        r=sc.nextDouble();
        System.out.print("Enter slant height of cone : ");
        sl_ht=sc.nextDouble();
        System.out.print("Enter height of cone : ");
        ht=sc.nextDouble();
        sur_ar=(pi*r*sl_ht)+(pi*Math.pow(r,2));
        vol=pi*Math.pow(r,2)*ht/3.0;
        System.out.println("Surface Area "+sur_ar);
        System.out.println("Volume "+vol);
    }
}
//Output
Enter radius of cone : 4
Enter slant height of cone : 6
Enter height of cone : 2.5
Surface Area 125.71428571428572
Volume 41.904761904761905
```

2. import java.util.*;

```
class chapter7_2
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        double l, g=9.8,T,pi=22.0/7.0;
        System.out.print("Enter the length of a pendulum : ");
        l=sc.nextDouble();
        T=2*pi*Math.sqrt(l)/g;
        System.out.println("Time period of the pendulum :"+T);
    }
}
//Output
Enter the length of a pendulum : 2
Time period of the pendulum :0.907075754291781
```

3. import java.util.*;

```
class chapter7_3
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        double a,b,c,r;
        System.out.print("Enter a : ");
        a=sc.nextDouble();
        System.out.print("Enter b : ");
        b=sc.nextDouble();
        System.out.print("Enter c : ");
        c=sc.nextDouble();
        r=Math.cbrt(a)+Math.pow(b,2)-Math.cbrt(c);
        System.out.print("Result : "+r);
    }
}
//Output
Enter a : 3
Enter b : 4
Enter c : 5
Result : 15.73227362363071
```

4. import java.util.*;

```
class chapter7_4
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        double d,a,c;
        System.out.print("Enter diameter : ");
        d=sc.nextDouble();
        a=22.0/7.0*Math.pow(d/2.0,2);
        c=2*22.0/7.0*d/2.0;
        System.out.println("Area : "+a);
        System.out.println("Circumference : "+c);
    }
}
//Output
Enter diameter : 50
Area : 1964.2857142857142
Circumference : 157.14285714285714
```

5. import java.util.*;

```
class chapter7_5
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        double p,e=2.71828,r,t,total_pl;
        System.out.print("Starting Population : ");
        p=sc.nextDouble();
        System.out.print("Percentage rate of population : ");
        r=sc.nextDouble();
        System.out.print("Time in year : ");
        t=sc.nextDouble();
        total_pl=p+Math.pow(e,(r*t));
        System.out.println("Total Population : "+total_pl);
    }
}
```

```
//Output
Starting Population : 10000
Percentage rate of population : 5.5
Time in year : 4
Total Population : 3.584869795693989E9
```

8. Conditional Constructs in Java

Unsolved Questions

A. 1. a 2. b 3. b 4. b 5. c 6. c
7. c 8. c

B. 1. Conditional operator 2. Control flow 3. After 4. boolean

C. a

D. 1. a. import java.util.*;

```
class chapter8_p1
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        double price,discount,finalPrice;
        // Input the price of the pen
        System.out.print("Enter the price of the pen: ");
        price = sc.nextDouble();
        // Condition to check for gift or discount
        if (price < 500)
        {
            System.out.println("You will receive a gift!");
        }
        else
        {
            discount = price * 0.25;
            finalPrice = price - discount;
            System.out.println("You get a 25% discount.");
         System.out.println("Discounted price: Rs. " + finalPrice);
```

```
        }
    }
}
//Output
Enter the price of the pen: 500
You get a 25% discount.
Discounted price: Rs. 375.0
```

b. import java.util.*;

```
class chapter8_p2
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num;
        // Input a positive number
        System.out.print("Enter a positive number: ");
        num = sc.nextInt();
        // Check if the number is positive
        if (num < 0)
        {
           System.out.println("Please enter a positive number.");
        }
        else
        {
            // Check if the number is even or odd
            if (num % 2 == 0)
            {
                System.out.println(num + " is an Even number.");
            }
            else
            {
                System.out.println(num + " is an Odd number.");
            }
        }
    }
}
//Output
```

```
Enter a positive number: 58
58 is an Even number.
```

c. import java.util.*;

```
class program8_p3
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int a1, a2,a3;
        // Input three angles of a triangle
        System.out.print("Enter first angle: ");
        a1 = sc.nextInt();
        System.out.print("Enter second angle: ");
        a2 = sc.nextInt();
        System.out.print("Enter third angle: ");
        a3 = sc.nextInt();
        // Check if the sum of angles is 180 degrees
        if (a1 + a2 + a3 == 180)
        {
            System.out.println("It is a valid triangle.");
        }
        else
        {
            System.out.println("Not a valid triangle. The angles
            must sum to 180 degrees.");
        }
   }
}
//Output
Enter first angle: 20
Enter second angle: 60
Enter third angle: 100
It is a valid triangle.
```

d. import java.util.*;

```
class program8_p4
 {
    public static void main() {
```

```
        Scanner sc = new Scanner(System.in);
        int day;
        // Input a number between 1 and 7
        System.out.print("Enter a number (1-7): ");
        day = sc.nextInt();

        // Check and display the corresponding weekday
        switch (day) {
            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! Please enter
                a number between 1 and 7.");
        }

    }
}
//Output
```

```
Enter a number (1-7): 4
Wednesday
```

e. import java.util.*;

```
class chapter8_p5
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num1,num2;
        // Accept two numbers
        System.out.print("Enter the first number: ");
        num1 = sc.nextInt();
        System.out.print("Enter the second number: ");
        num2 = sc.nextInt();
        // Compare numbers
        if (num1 > num2)
        {
            System.out.println("Square of the greater number ("
            + num1 + "): " + (num1 * num1));
            System.out.println("Cube of the smaller number (" +
            num2 + "): " + (num2 * num2 * num2));
        }
        else if (num2 > num1)
        {
            System.out.println("Square of the greater number ("
            + num2 + "): " + (num2 * num2));
            System.out.println("Cube of the smaller number (" +
            num1 + "): " + (num1 * num1 * num1));
        }
        else
        {
            System.out.println("Both numbers are equal. Twice of
            the number (" + num1 + "): " + (2 * num1));
        }
    }
}
//Output
```

```
Enter the first number: 4
Enter the second number: 6
Square of the greater number (6): 36
Cube of the smaller number (4): 64
```

f. import java.util.*;

```
class chapter8_p6
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String name;
        double amt,dis_rate,dis_Amt,fPrice;
        // Input customer's name
        System.out.print("Enter customer's name: ");
        name = sc.nextLine();
        // Input amount of purchase
        System.out.print("Enter amount of purchase: Rs. ");
        amt = sc.nextDouble();
        // Determine discount rate based on purchase amount
        if (amt <= 3000)
        {
            dis_rate = 5;
        }
        else if (amt < 5000)
        {
            dis_rate = 7;
        }
        else
        {
            dis_rate = 10;
        }
        // Calculate discount and final price
        dis_Amt = (dis_rate / 100) * amt;
        fPrice = amt - dis_Amt;
        // Display results
      System.out.println("-------------------------------------");
```

```
            System.out.println("\tBill Amount");
            System.out.println("Customer Name: " + name);
            System.out.println("Total Purchase Amount: Rs. " + amt);
           System.out.println("Discount Applied: " + dis_rate + "%");
            System.out.println("Discount Amount: Rs. " + dis_Amt);
            System.out.println("Final Price After Discount: Rs. " +
            fPrice);
        }
    }
    //Output
    Enter customer's name: Amit Sen
    Enter amount of purchase: Rs. 2000
    -----------------------------------
    Bill Amount
    Customer Name: Amit Sen
    Total Purchase Amount: Rs. 2000.0
    Discount Applied: 5.0%
    Discount Amount: Rs. 100.0
    Final Price After Discount: Rs. 1900.0
```

g. import java.util.*;

```
    class chapter8_p7
    {
        public static void main() {
            Scanner sc = new Scanner(System.in);
            double bill_amt,f_amt=0.0,dis;
            String p_Mode,facility="No discount";
            // Input bill amount
            System.out.print("Enter the bill amount: Rs. ");
            bill_amt = sc.nextDouble();

            // Input mode of payment
           System.out.print("Enter mode of payment (cc/dc/ew/c): ");
            p_Mode = sc.next().toLowerCase(); // Convert to lowercase
            for easy comparison
            // Apply discount or cashback based on payment mode
            switch (p_Mode) {
                case "cc": // Credit Card
```

```
                dis = bill_amt * 0.015;
                f_amt = bill_amt - dis;
                facility = "1.5% discount applied (Rs. " + dis +
                ")";
                break;
            case "dc": // Debit Card
                f_amt = bill_amt - 10;
                facility = "Rs. 10 cash back";
                break;
            case "ew": // E-Wallet
                f_amt = bill_amt - 20;
                facility = "Rs.20 cash back";
                break;
            case "c": // Cash
                facility = "No discount";
                break;
            default:
                System.out.println("Invalid payment mode!");

        }

        // Display results
      System.out.println("\nPayment Mode: " + p_Mode.toUpperCase());
        System.out.println("Facility: " + facility);
        System.out.println("Final Amount to be Paid: Rs. " + f_
        amt);
    }
}
//Output
Enter the bill amount: Rs. 1000
Enter mode of payment (cc/dc/ew/c): cc

Payment Mode: CC
Facility: 1.5% discount applied (Rs. 15.0)
Final Amount to be Paid: Rs. 985.0
```

h. import java.util.*;

```
class chapter8_p8
```

```
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int calls;
        double bill_amt;
        // Input the number of calls
        System.out.print("Enter the number of calls made: ");
        calls = sc.nextInt();
        bill_amt = 100; // Minimum charge for up to 50 calls

        // Calculate bill based on the number of calls
        if (calls > 50)
        {
            if (calls <= 100)
            {
                bill_amt += (calls - 50) * 0.80;
            }
            else if (calls <= 200)
            {
               bill_amt += (50 * 0.80) + ((calls - 100) * 0.60);
            }
            else
            {
                bill_amt += (50 * 0.80) + (100 * 0.60) + ((calls
                - 200) * 0.40);
            }
        }

        // Display the final bill amount
        System.out.println("Total Telephone Bill: Rs.  "+ bill_
        amt);
    }
}
//Output
Enter the number of calls made: 150
Total Telephone Bill: Rs.  170.0
```

i. import java.util.*;

```
class chapter8_p9
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int sales;
        double s_amt,c_rate;
        String gift;
        // Input number of products sold
        System.out.print("Enter the number of products sold: ");
        sales = sc.nextInt();
        // Input total selling amount
        System.out.print("Enter the total selling amount:Rs. ");
        s_amt = sc.nextDouble();
        // Determine commission and gift based on sales count
        if (sales <= 50)
        {
            c_rate = 5;
            gift = "A Parker pen";
        }
        else if (sales <= 75)
        {
            c_rate = 7.5;
            gift = "A Micro SD card";
        }
        else if (sales <= 100)
        {
            c_rate = 10;
            gift = "A Mobile";
        }
        else
        {
            c_rate = 15;
            gift = "A Laptop";
        }
        // Calculate commission
```

```
            double com = (c_rate / 100) * s_amt;
            // Display results
          System.out.println("\nSalesperson's Performance Summary:");
           System.out.println("Total Sales: " + sales + " products");
            System.out.println("Total Selling Amount: ₹" + s_amt);
            System.out.println("Commission Earned (" + c_rate + "%):
            Rs. " + com);
            System.out.println("Gift Received: " + gift);
        }
    }
    //Output
    Enter the number of products sold: 40
    Enter the total selling amount:Rs. 2000

    Salesperson's Performance Summary:
    Total Sales: 40 products
    Total Selling Amount: ₹2000.0
    Commission Earned (5.0%): Rs. 100.0
    Gift Received: A Parker pen
```

j. import java.util.*;

```
    class chapter8_p10
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            double inc,tax=0.0;
            // Input annual taxable income
           System.out.print("Enter the annual taxable income: Rs. ");
            inc = sc.nextDouble();
            // Compute tax based on income slabs
            if (inc <= 100000)
            {
                tax = 0; // No tax
            }
            else if (inc <= 150000)
            {
```

```
            tax = (inc - 100000) * 0.10; // 10% for income exceeding
            1,00,000
        }
        else if (inc <= 250000)
        {
            tax = 5000 + (inc - 150000) * 0.20; // Rs. 5000 + 20%
            for income exceeding 1,50,000
        } else
        {
            tax = 25000 + (inc - 250000) * 0.30; // Rs. 25000 +
            30% for income exceeding 2,50,000
        }
        // Display the calculated tax
        System.out.println("Total Income Tax to be Paid: Rs. " +
        tax);
    }
}
//Output
Enter the annual taxable income: Rs. 150000
Total Income Tax to be Paid: Rs. 5000.0
```

k. import java.util.*;

```
class chapter8_p11
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        char ch;
        // Input a character
        System.out.print("Enter a character: ");
        ch = sc.next().toLowerCase().charAt(0); // Convert to
        lowercase for easy comparison
        // Check if the character is a vowel or consonant
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' ||
        ch == 'u')
        {
            System.out.println("The character '" + ch + "' is a
            Vowel.");
```

```
            }
            else if ((ch >= 'a' && ch <= 'z'))
            {
                System.out.println("The character '" + ch + "' is a
                Consonant.");
            }
            else
            {
                System.out.println("Invalid input! Please enter an
                alphabet.");
            }
        }
    }
    //Output
    Enter a character: A
    The character 'a' is a Vowel.
```

2. a. import java.util.*;

```
    class chapter8_p12
     {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            double PI = 22.0/7.0,r,ht,vol;
            int ch;

            // Menu
            System.out.println("Volume Calculation Menu:");
            System.out.println("1. Volume of Sphere");
            System.out.println("2. Volume of Cylinder");
            System.out.println("3. Volume of Cone");
            System.out.print("Enter your choice (1-3): ");
            ch = sc.nextInt();
            switch (ch) {
                case 1:
                    // Volume of Sphere: (4/3)πr³
                    System.out.print("Enter the radius of the sphere:
                    ");
```

```
                r = sc.nextDouble();
                vol = (4.0 / 3) * PI * Math.pow(r, 3);
              System.out.println("Volume of the Sphere: " + vol);
                break;

            case 2:
                // Volume of Cylinder: πr²h
                System.out.print("Enter the radius of the cylinder:
                ");
                r = sc.nextDouble();
                System.out.print("Enter the height of the cylinder:
                ");
                ht = sc.nextDouble();
                vol = PI * Math.pow(r, 2) * ht;
                System.out.println("Volume of the Cylinder: " +
                vol);
                break;

            case 3:
                // Volume of Cone: (πr²h) / 3
                System.out.print("Enter the radius of the cone:
                ");
                r = sc.nextDouble();
                System.out.print("Enter the height of the cone:
                ");
                ht = sc.nextDouble();
                vol = (PI * Math.pow(r, 2) * ht) / 3.0;
               System.out.println("Volume of the Cone: " + vol);
                break;

            default:
                System.out.println("Invalid choice! Please enter
                a valid option (1-3).");
        }

    }
}
```

```
//Output
Volume Calculation Menu:
1. Volume of Sphere
2. Volume of Cylinder
3. Volume of Cone
Enter your choice (1-3): 1
Enter the radius of the sphere: 2
Volume of the Sphere: 33.52380952380952
```

b. import java.util.*;

```
class Mchapter8_p13
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Display the menu
        double fahren,cel;
        int ch,n;
        System.out.println("Menu:");
        System.out.println("1. Convert Fahrenheit to Celsius");
        System.out.println("2. Convert Celsius to Fahrenheit");
        System.out.println("3. Check if a number is Even or
        Odd");
        System.out.print("Enter your choice (1-3): ");
        ch = sc.nextInt();
        switch (ch)
        {
            case 1:
                // Fahrenheit to Celsius: C = (F - 32) * 5/9
                System.out.print("Enter temperature in Fahrenheit:
                ");
                fahren = sc.nextDouble();
                cel = (fahren - 32) * 5 / 9.0;
                System.out.println("Temperature in Celsius: " +
                cel + "°C");
                break;
            case 2:
                // Celsius to Fahrenheit: F = (C * 9/5) + 32
```

```
                System.out.print("Enter temperature in Celsius:
                ");
                cel = sc.nextDouble();
                fahren = (cel * 9 / 5.0) + 32;
                System.out.println("Temperature in Fahrenheit: "
                + fahren + "°F");
                break;
            case 3:
                // Check if number is Even or Odd
                System.out.print("Enter a number: ");
                n = sc.nextInt();
                if (n % 2 == 0)
                {
                    System.out.println("The number " + n + " is
                    Even.");
                }
                else
                {
                    System.out.println("The number " + n + " is
                    Odd.");
                }
                break;
            default:
                System.out.println("Invalid choice! Please enter
                a valid option (1-3).");
        }
    }
}
//Output
Menu:
1. Convert Fahrenheit to Celsius
2. Convert Celsius to Fahrenheit
3. Check if a number is Even or Odd
Enter your choice (1-3): 1
Enter temperature in Fahrenheit: 20
Temperature in Celsius: -6.67°C
```

c. import java.util.*;

```
class chapter8_p14
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch,num,i,counter=0,f=1;
        // Display the menu
        System.out.println("Menu:");
        System.out.println("1. Check if a number is Prime");
        System.out.println("2. Find the Factorial of a number");
        System.out.print("Enter your choice (1-2): ");
        ch = sc.nextInt();

        switch (ch)
        {
            case 1:
                // Prime number check
                System.out.print("Enter a number to check if it
                is Prime: ");
                num = sc.nextInt();
                for(i=2;i<num;i++)
                {
                    if (num%i==0)
                        counter++;
                }
                if(counter==0)
                {
                 System.out.println(num + " is a Prime number.");
                }
                else
                {
                    System.out.println(num + " is NOT a Prime
                    number.");
                }
                break;
```

```
                case 2:
                    // Factorial calculation
                    System.out.print("Enter a number to find its
                    Factorial: ");
                    int n = sc.nextInt();
                    for(i=1;i<=n;i++)
                    {
                        f=f*i;
                    }
                    if (n < 0)
                    {
                        System.out.println("Factorial is not defined
                        for negative numbers.");
                    }
                    else
                    {
                        System.out.println("Factorial of " + n + "
                        is: " + f);
                    }
                    break;
                default:
                    System.out.println("Invalid choice! Please enter
                    1 or 2.");
            }

        }

    }
    //Output
    Menu:
    1. Check if a number is Prime
    2. Find the Factorial of a number
    Enter your choice (1-2): 1
    Enter a number to check if it is Prime: 37
    37 is a Prime number.
```

d. import java.util.*;

```
class chapter8_p15
```

```
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch;
        // Display the menu
        System.out.println("Menu:");
        System.out.println("1. Square root of 9");
        System.out.println("2. Absolute value of 126.4");
        System.out.println("3. Smallest integer greater than 56.7");
        System.out.println("4. Print a random value between 0 and
        1");
        System.out.print("Enter your choice (1-4): ");
        ch = sc.nextInt();
        switch (ch)
        {
            case 1:
                // Square root of 9
                System.out.println("Square root of 9: " + Math.
                sqrt(9));
                break;
            case 2:
                // Absolute value of 126.4
                System.out.println("Absolute value of 126.4: " +
                Math.abs(126.4));
                break;
            case 3:
                // Smallest integer greater than 56.7 (Ceiling
                value)
                System.out.println("Smallest integer greater than
                56.7: " + Math.ceil(56.7));
                break;
            case 4:
                // Random value between 0 and 1
                System.out.println("Random value between 0 and 1:
                " + Math.random());
                break;
```

```
                default:
                    System.out.println("Invalid choice! Please enter
                    a valid option (1-4).");
            }
        }
    }
    //Output
    Menu:
    1. Square root of 9
    2. Absolute value of 126.4
    3. Smallest integer greater than 56.7
    4. Print a random value between 0 and 1
    Enter your choice (1-4): 1
    Square root of 9: 3.0
```

e. import java.util.*;

```
class chapter8_p16
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Display menu
        int ch,len,bre,ar1,ar2,s;
        double r,ar3;
        System.out.println("Menu:");
        System.out.println("1. Area of a Rectangle");
        System.out.println("2. Area of a Square");
        System.out.println("3. Area of a Circle");
        System.out.print("Enter your choice (1-3): ");
        ch = sc.nextInt();

        if (ch == 1)
        {
            // Area of Rectangle: length × breadth
            System.out.print("Enter length of the rectangle: ");
            len = sc.nextInt();
           System.out.print("Enter breadth of the rectangle: ");
            bre = sc.nextInt();
```

```
            ar1 = len * bre;
            System.out.println("Area of the rectangle: " + ar1);
        }
        else if (ch == 2)
        {
            // Area of Square: side × side
            System.out.print("Enter side of the square: ");
            s = sc.nextInt();
            ar2 = s * s;
            System.out.println("Area of the square: " + ar2);
        }
        else if (ch == 3)
        {
            // Area of Circle: π × radius²
            System.out.print("Enter radius of the circle: ");
            r = sc.nextDouble();
            ar3 = 22.0/7.0 * r * r;
            System.out.println("Area of the circle: " + ar3);
        }
        else
        {
            System.out.println("Invalid choice!");
        }
    }
}
//Output
Menu:
1. Area of a Rectangle
2. Area of a Square
3. Area of a Circle
Enter your choice (1-3): 1
Enter length of the rectangle: 6
Enter breadth of the rectangle: 3
Area of the rectangle: 18
```

9. Iterative Constructs in Java

Unsolved Questions

A. 1. d 2. c 3. c 4. c

B. 1. Null 2. break, continue 3. for(i=3;i<=30;i=i+3)
4. zero 5. entry 6. while

C. 1. Different parts of loops are:

a. Initialisation: In this part, a variable is initialized that helps the loop to execute.

b. Condition for testing: The loop will stop executing when the condition changes its boolean value. i.e. true or false.

c. Increment or decrement: Each and every iteration will increase the value of the variable by a number.

d. Body of the loop: Every loop has statements that the user wants to repeat to perform the desired task.

2. The two categories of loops are: Entry control loop and Exit control loop

3. **Infinite Loop:** An infinite loop continues running forever unless a break statement, exception, or external force (e.g., stopping the program) terminates it.

Example: Infinite while Loop

```
class InfiniteLoop {
    public static void main() {
        while (true) {  // No condition to exit
            System.out.println("This is an infinite loop!");
        }
    }
}
```

Finite Loop: A finite loop runs for a fixed number of iterations or until a condition is met.

Example: Finite for Loop

```
class FiniteForLoop {
    public static void main() {
        for (int i = 0; i < 5; i++) {  // Runs 5 times
            System.out.println("Iteration: " + (i + 1));
        }
    }
}
```

4. A delay loop is a type of loop that runs for a specific duration to introduce a delay in program execution.

   Example:

```
for (int i = 0; i<10000; i++)
        {
            // this will execute without any output
        }
```

5. Interconversion of loop:

   Loops in Java can be converted from for → while → do-while without changing their functionality.

   1. Converting for loop to while loop

      Example: for loop

```
class ForLoop {
    public static void main() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Iteration: " + i);
        }
    }
}
```

      Equivalent while loop

```
class WhileLoop {
    public static void main() {
        int i = 1;
        while (i <= 5) {
            System.out.println("Iteration: " + i);
            i++;
        }
    }
}
```

   2. Converting while loop to do-while loop

      Example: while loop

```
class WhileLoop {
    public static void main() {
        int i = 1;
        while (i <= 5) {
```

```
            System.out.println("Iteration: " + i);
            i++;
        }
    }
}
```

Equivalent do-while loop

```
class DoWhileLoop {
    public static void main() {
        int i = 1;
        do {
            System.out.println("Iteration: " + i);
            i++;
        } while (i <= 5);
    }
}
```

3. Converting do-while loop to for loop

Example: do-while loop

```
class DoWhileLoop {
    public static void main() {
        int i = 1;
        do {
            System.out.println("Iteration: " + i);
            i++;
        } while (i <= 5);
    }
}
```

Equivalent for loop

```
class ForLoop {
    public static void main () {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Iteration: " + i);
        }
    }
}
```

D. d

E. 1. Write the programs in Java to display the following series:

a. 1,3,5,7,9,...99

```
class chapter9_PE1
{
    public static void main ()
    {
        for (int i = 1; i <= 99; i += 2)
        {
            System.out.print(i + " ");
        }
    }
}
```

b. 20,18,16,....2

```
class chapter9_PE2
{
    public static void main () {

        for (int i = 20; i >= 2; i -= 2) {
            System.out.print(i + " ");
        }

    }
```

c. 2,4,8,16,.....,256

```
class chapter9_PE3
{
public static void main () {
        System.out.println("Using while loop:");
        int i = 2;
        while (i <= 256) {
            System.out.print(i + " ");
            i *= 2;
        }
        System.out.println();
    }
}
```

d. 1/3,2/6,3/9,....10/30

```
class chapter9_PE4
{
public static void main () {
        System.out.println("Using for loop:");
        for (int i = 1; i <= 10; i++) {
            System.out.println(i + "/" + (i * 3) + " ");
        }
        System.out.println();
    }
}
```

e. 1,12,123,1234,12345

```
class chapter9_PE5
{
public static void main () {
        String num = "";
        for (int i = 1; i <= 5; i++) {
            num += i;
            System.out.println(num);
        }
    }
}
```

f. 1,11,111,1111,11111,111111

```
class NumberPattern
{
public static void main()
{
       int num = 1;
       for (int i = 1; i <= 6; i++)
        {
           System.out.println(num);
           num = num * 10 + 1;
        }
    }
}
```

2. Write the programs in Java to calculate the result of the following series:

a. s=1+4+9+16+25

```
class ForLoop
{
     public static void main() {

         int sum = 0;
         for (int i = 1; i <= 5; i++) {
             sum += i * i;
         }
         System.out.println("Sum = " + sum);
     }
}
```

b. p=10x9x8x7x6x5x4x3x2x1

```
class ForLoop
{
     public static void mainb() {
         int fact= 1;
         for (int i = 10; i >= 1; i--) {
             fact *= i;
         }
         System.out.println("Factorial of 10 = " + fact);
     }
}
```

c. s=1+11+111+1111+11111

```
class ForLoop
{
     public static void main() {

         int sum = 0,num = 1,i;
         for (i = 1; i <= 5; i++) {
             sum += num;
             num = num * 10 + 1;
         }
         System.out.println("Sum = " + sum);
     }
}
```

d. s=10+20+30+40+50+...+100

```
class ForLoop
{
      public static void main()
      {
        int sum = 0,i;
        for (i = 10; i <= 100; i += 10)
        {
            sum += i;
        }
        System.out.println("Sum = " + sum);
    }
}
```

e. p=1/2*2/3*3/4*...*9/10

```
class ForLoop
{
    public static void main() {
        double pro = 1.0,i;
        for (i = 1; i <= 9; i++) {
            pro *= (double) i / (i + 1);
        }
        System.out.println("Pro = " + pro);
    }
}
```

f. s=2+5+10+17+...+nth term

```
class ForLoop
{
    public static void main(int n) {
        int sum = 0,term = 2,i;
        for (i = 1; i <= n; i++) {
            sum += term;
            term += (2 * i + 1);
        }
        System.out.println("Sum = " + sum);
    }
}
```

3. import java.util.*;

```
class Neon_Number
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num,sq,sum=0,rem;
        // Taking user input
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        // Calculate square of the number
        sq = num * num;
        // Calculate sum of digits of the square
        while (sq > 0) {
            rem= sq % 10;  // Extract last digit and add to sum
            sum += rem;
            sq /= 10;         // Remove last digit
        }
       // Check if the sum of digits is equal to the original number
        if (sum == num)
        {
            System.out.println(num + " is a Neon Number.");
        }
        else
        {
            System.out.println(num + " is NOT a Neon Number.");
        }
    }
}
//Output
Enter a number: 9
9 is a Neon Number.
```

4. import java.util.*;

```
class Digit_Separator
{
    public static void main()
    {
```

```
        Scanner sc = new Scanner(System.in);
        int num, rem;
        // Taking user input
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        while (num > 0)
        {
            rem= num % 10;  // Extract last digit and add to sum
            System.out.print(rem+",");
            num /= 10;        // Remove last digit
        }

    }
}
//Output
Enter a number: 43251
1,5,2,3,4,
```

5. import java.util.*;

```
class absolute_diff
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num,rev=0,rem,diff,temp;
        // Taking user input
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        temp = num;
        System.out.println("Original Number "+ num);
        while (num > 0)
        {
            rem= num % 10;  // Extract last digit and add to sum
            rev=rev*10+rem;
            num /= 10;        // Remove last digit
        }
        diff = Math.abs(temp-rev);
        System.out.println("Reversed Number "+ rev);
```

```
            System.out.println("Absolute Difference "+ diff);

        }
    }
    //Output
    Enter a number: 123
    Original Number 123
    Reversed Number 321
    Absolute Difference 198
```

6. import java.util.*;

```
    class Disarium_Number
    {
        // Function to check if a number is a Disarium Number
        public static void main(int num)
        {
            int sum = 0, temp = num, count = 0,rem;
            while (temp > 0)
            {
                count++;
                temp /= 10; // Remove last digit
            }
            temp = num;
            while (temp > 0)
            {
                rem = temp % 10; // Extract last digit
                sum += Math.pow(rem, count); // Raise to position power
                temp /= 10; // Remove last digit
                count--; // Decrease position count
            }
            // Checking for Disarium Number
            if (sum==num) {
                System.out.println(num + " is a Disarium Number.");
            } else {
                System.out.println(num + " is NOT a Disarium Number.");
            }
        }
    }
```

```
//Output
135 is a Disarium Number.
```

7. a. import java.util.*;

```
class Series1
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int a,n,i;
        long sum = 0;
        // Taking input for base 'a' and exponent limit 'n'
        System.out.print("Enter the base (a): ");
        a = sc.nextInt();
        System.out.print("Enter the exponent limit (n): ");
        n = sc.nextInt();
        // Calculating the sum of the series
        for (i = 1; i <= n; i++)
        {
            sum += Math.pow(a, i); // a^i and add to sum
        }
        // Display the result
        System.out.println("Sum of the series: " + sum);
    }
}
//Output
Enter the base (a): 4
Enter the exponent limit (n): 3
Sum of the series: 84
```

b. import java.util.*;

```
class Series2
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int x,n,i;
        long sum = 0;
        // Taking input for base 'a' and exponent limit 'n'
```

```
            System.out.print("Enter the base (x): ");
            x = sc.nextInt();
            System.out.print("Enter the n: ");
            n = sc.nextInt();
            // Calculating the sum of the series
            for (i = 1; i <= n; i++)
            {
                sum += i*Math.pow(x, 2); // a^i and add to sum
            }
            // Display the result
            System.out.println("Sum of the series: " + sum);
        }
    }
    //Output
    Enter the base (x): 6
    Enter the n: 4
    Sum of the series: 360
```

c. import java.util.*;

```
    class Series3
     {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            // Taking input for the number of terms (n)
            System.out.print("Enter the number of terms (n): ");
            int n = sc.nextInt(),i;
            // Variable to store the product
            long pro = 1;
            for (i = 1; i <= n; i++) {
                pro *= Math.pow(i, i); // Multiply i^i to product
            }

            // Display the result
            System.out.println("Product of the series: " + pro);
        }
    }
    //Output
```

```
Enter the number of terms (n): 4
Product of the series: 27648
```

d. import java.util.*;

```
class Series4
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int sum = 0,i,term=1,n;
        System.out.println("Enter n : ");
        n=sc.nextInt();
        for (i = 1; i <= n; i++)
        {
            sum += term; // Add the term to sum
            term += i;   // Generate the next term
        }

        // Display the result
        System.out.println("Sum of the series: " + sum);
    }
}
//Output
Enter n :
10
Sum of the series: 175
```

e. class Series4

```
 {
    public static void main()
    {
        int i,f=1;
        double sum = 0.0;
        // Calculate the sum of the series
        for (i = 2; i <= 10; i += 2)
        {
            f=f*(i-1)*i;
            sum += (double) i / f;
        }
```

```
            // Display the result
            System.out.println("Sum of the series: " + sum);
        }
    }
    //Output
    Sum of the series: 1.1752011684303352
```

f. import java.util.*;

```
    class Series5
     {
        public static void main()
        {
            Scanner sc= new Scanner(System.in);
            System.out.print("Enter the value of a: ");
            double a = sc.nextDouble(), sum = 0.0;
            int i;

            // Calculate the series sum for sqrt(2) to sqrt(10) with
            even numbers
            for (i = 2; i <= 10; i += 2)
            {
                sum += a / Math.sqrt(i); // a / sqrt(i)
            }
            System.out.println(sum);
        }
    }
    //Output
    Enter the value of a: 10
    22.851362282605223
```

g. import java.util.*;

```
    class Series6
     {
        public static void main()
        {
            Scanner sc= new Scanner(System.in);
            int i,a,n,term;
            long sum = 0;
            // Taking input for base 'a' and exponent limit 'n'
```

```
            System.out.print("Enter the base (a): ");
            a = sc.nextInt();
            System.out.print("Enter the exponent limit (n): ");
            n = sc.nextInt();
            // Calculate the sum of the series

            for (i = 1; i <= n; i++) {
                term = (int) Math.pow(a, i); // a^i
                if (i % 2 == 0) {
                    sum -= term; // Subtract even exponent terms
                } else {
                    sum += term; // Add odd exponent terms
                }
            }
            System.out.println("The sum of the series "+sum);
        }
    }
    //Output
    Enter the base (a): 5
    Enter the exponent limit (n): 4
    The sum of the series -520
```

h. import java.util.*;

```
    class Series7
     {
        public static void main()
        {
            Scanner sc= new Scanner(System.in);
            // Taking input for base 'x' and limit 'n'
            int n,x,i;
            long pro = 1;
            System.out.print("Enter the value of x: ");
            x = sc.nextInt();
            System.out.print("Enter the value of n: ");
            n = sc.nextInt();
            // Loop to calculate the product
            for (i = 1; i <= n; i += 2)
            {
```

```
                pro *= (x + i); // Multiply each term
            }

            // Display the result
            System.out.println("Product of the series: " + pro);
        }
    }
    //Output
    Enter the value of x: 5
    Enter the value of n: 3
    Product of the series: 48
```

8. import java.util.*;

```
class chapter9_prog8
 {
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int num,sumOfDigits = 0, temp;
        // Input number
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        temp = num;
        // Calculate sum of digits
        while (temp > 0)
        {
            sumOfDigits += temp % 10;
            temp /= 10;
        }
        // Check divisibility
        if (num % sumOfDigits == 0)
        {
            System.out.println(num + " is a Niven Number.");
        }
        else
        {
            System.out.println(num + " is NOT a Niven Number.");
        }
```

```
    }
}
//Output
Enter a number: 111
111 is a Niven Number.
```

9.
```
import java.util.*;
class chapter9_prog8
 {
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt(),temp,c=0,rem;
        long square = (long) num * num;
        temp = num;
        boolean f=true;
        // Check if the last digits of square match num
        while (temp > 0)
        {
            c++;
            temp /= 10;
        }

        if (num==square%(int)Math.pow(10,c))
        {
           System.out.println(num + " is an Automorphic Number.");
        }
        else
        {
         System.out.println(num + " is NOT an Automorphic Number.");
        }

    }
}
//Output
Enter a number: 25
25 is an Automorphic Number.
```

10. import java.util.*;

```
class Series
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch,n,sum=0,num=0,i;
        double product = 1.0;
        // Display menu
        System.out.println("\nMENU:");
        System.out.println("1. Compute sum of series: S = 1 + 12 +
        123 + 1234 + 12345");
        System.out.println("2. Compute product of series: P = (1/1)
        * (2/4) * (3/9) * ... * (n/n^2)");
        System.out.print("Enter your choice: ");
        ch = sc.nextInt();
        switch (ch)
        {
            case 1:
                for (i = 1; i <= 5; i++)
                {
                    num = num * 10 + i;
                    sum += num;
                }
                System.out.println("Sum of the series: " + sum);
                break;
            case 2:
               System.out.print("Enter the number of terms (n): ");
                n = sc.nextInt();
                for (i = 1; i <= n; i++)
                {
                    product *= (double) i / (i * i);
                }
                System.out.println("Product of the series: " +
                product);
                break;
            default:
```

```
                System.out.println("Invalid choice! Please select
                again.");
            }
        }
    }
    //Output
    MENU:
    1.Compute sum of series: S = 1 + 12 + 123 + 1234 + 12345
    2.Compute product of series: P = (1/1) * (2/4) * (3/9) * ... * (n/
      n^2)
      Enter your choice: 1
      Sum of the series: 13715
```

11. import java.util.*;

```
class Series_Switch {
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int ch,n,i,j;
        double sum = 0,x,term;
        long fact;
        // Display menu
        System.out.println("\nMENU:");
        System.out.println("1. Compute sum of series: S = X + X^2/2!
        + X^3/3! + ... + X^n/n!");
        System.out.println("2. Compute sum of alternating series: S
        = 1/1^3 - 1/2^3 + 1/3^3 - ... ± 1/n^3");
        System.out.print("Enter your choice: ");
        ch = sc.nextInt();
        switch (ch)
        {
            case 1:
                System.out.print("Enter the value of X: ");
                x = sc.nextDouble();
               System.out.print("Enter the number of terms (n): ");
                n = sc.nextInt();
                for (i = 1; i <= n; i++)
                {
                    fact = 1;
```

```
                    for (j = 1; j <= i; j++)
                    {
                        fact *= j;
                    }
                    sum += Math.pow(x, i) / fact;
                }
                System.out.println("Sum of the series: " + sum);
                break;
            case 2:
               System.out.print("Enter the number of terms (n): ");
                n = sc.nextInt();
                for (i = 1; i <= n; i++)
                {
                    term = 1.0 / Math.pow(i, 3);
                    if (i % 2 == 0)
                    {
                        sum -= term; // Subtract for even i
                    }
                    else
                    {
                        sum += term; // Add for odd i
                    }
                }
                System.out.println("Sum of the alternating series:
                " + sum);
                break;
            default:
                System.out.println("Invalid choice! Please select
                again.");
        }

    }
}
//Output
MENU:
1.Compute sum of series: S = X + X^2/2! + X^3/3! + ... + X^n/n!
```

2. Compute sum of alternating series: S = 1/1^3 - 1/2^3 + 1/3^3 - ... ± 1/n^3
Enter your choice: 1
Enter the value of X: 3
Enter the number of terms (n): 4
Sum of the series: 15.375

12. import java.util.*;

```
class Series_Program
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch,n,i;
        double sum;
        // Display menu
        System.out.println("\nMENU:");
        System.out.println("1. Generate sequence: 0, 3, 8, 15, 24,
        ... up to n terms");
        System.out.println("2. Compute sum of series: S = 1/2 + 3/4
        + 5/6 + ... + 9/10");
        System.out.println("3. Exit");
        System.out.print("Enter your choice: ");
        ch = sc.nextInt();
        if (ch == 1)
        {
            // Generating sequence 0, 3, 8, 15, 24, ...
            System.out.print("Enter the number of terms (n): ");
            n = sc.nextInt();
            System.out.print("Sequence: ");
            for (i = 1; i <= n; i++) {
                System.out.print(i*i - 1+ " ");
                // Formula to generate the next term
            }
            System.out.println();
        }
        else if(ch == 2)
        {
```

```
                // Computing sum of series: S = 1/2 + 3/4 + 5/6 + ... +
                9/10
                sum = 0;
                for (i = 1; i <= 9; i += 2)
                { // Odd numerators: 1, 3, 5, 7, 9
                    sum += (double) i / (i + 1); // Denominator is always
                    numerator + 1
                }
                System.out.println("Sum of the series: " + sum);
            }
            else
            {
                System.out.println("Invalid choice! ");
            }
        }
    }
    //Output
    MENU:
    1.Generate sequence: 0, 3, 8, 15, 24, ... up to n terms
    2.Compute sum of series: S = 1/2 + 3/4 + 5/6 + ... + 9/10
    3.Exit
      Enter your choice: 2
      Sum of the series: 3.8583333333333334
```

13. import java.util.*;

```
class DuckNumber {
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int num,temp,rem;
        // Input: Number as an integer
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        temp = num; // Store original number for output
        boolean isDuck = false;
        // Remove leading zeros
        while (num > 0 && num % 10 == 0) {
            num /= 10; // Remove trailing zeros if any
        }
```

```
            num=temp;
            // Extract digits and check for '0'
            while (num > 0) {
                rem = num % 10; // Extract last digit
                if (rem == 0) {
                    isDuck = true;
                    break;
                }
                num /= 10; // Remove last digit
            }

            // Print result
            if (isDuck) {
                System.out.println(temp + " is a Duck Number.");
            } else {
                System.out.println(temp + " is NOT a Duck Number.");
            }

        }
    }
    //Output
    Enter a number: 12045
    12045 is a Duck Number.
```

14. import java.util.*;

```
    class Factor_Factorial
     {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            int ch,num,i;
            // Display menu
            System.out.println("MENU:");
            System.out.println("1. Print the factors of a number");
            System.out.println("2. Print the factorial of a number");
            System.out.print("Enter your choice: ");
            ch = sc.nextInt();
            System.out.print("Enter a number: ");
```

```
            num = sc.nextInt();
            switch (ch) {
                case 1:
                    // Finding and printing factors
                    System.out.print("Factors of " + num + ": ");
                    for (i = 1; i <= num; i++) {
                        if (num % i == 0) {
                            System.out.print(i + " ");
                        }
                    }
                    System.out.println();
                    break;

                case 2:
                    // Finding and printing factorial
                    long factorial = 1;
                    for (i = 1; i <= num; i++) {
                        factorial *= i;
                    }
                    System.out.println("Factorial of " + num + " is: "
                    + factorial);
                    break;

                default:
                    System.out.println("Invalid choice! Please select 1
                    or 2.");
            }

        }
    }
    //Output
    MENU:
    1. Print the factors of a number
    2. Print the factorial of a number
      Enter your choice: 1
      Enter a number: 10
      Factors of 10: 1 2 5 10
```

15. import java.util.*;

```
class Smallest_Digit {
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int num, sm=9,temp,rem;
        // Input: Read the number
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        temp = num; // Store original number for display

        // Extract digits one by one
        while (num > 0) {
            rem = num % 10; // Get last digit

            // Update smallest digit if found smaller
            if (rem < sm) {
                sm = rem;
            }

            num /= 10; // Remove last digit
        }

        // Print the result
        System.out.println("The smallest digit in " + temp + " is:
        " + sm);

    }
}
//Output
Enter a number: 15375
The smallest digit in 15375 is: 1
```

16. import java.util.*;

```
class switch_case {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch,n;
```

```
// Display menu
System.out.println("MENU:");
System.out.println("1. Generate Fibonacci series up to nth
term");
System.out.println("2. Find product of even digits in a
number");
System.out.print("Enter your choice: ");
ch = sc.nextInt();

switch (ch) {
    case 1:
        // Fibonacci series generation
       System.out.print("Enter the number of terms (n): ");
        n = sc.nextInt();

        int a = 0, b = 1, c;
        System.out.print("Fibonacci Series: ");
        for (int i = 1; i <= n; i++) {
            System.out.print(a + " ");
            c = a + b;
            a = b;
            b = c;
        }
        System.out.println();
        break;

    case 2:
        // Product of even digits in a number
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int product = 1;
        boolean hasEven = false;

        while (num > 0) {
            int digit = num % 10; // Extract last digit
            if (digit % 2 == 0 && digit != 0) { // Check if
            even and not zero
```

```
                    product *= digit;
                    hasEven = true;
                }
                num /= 10; // Remove last digit
            }

            if (hasEven) {
                System.out.println("Product of even digits: " +
                product);
            } else {
                System.out.println("No even digits found.");
            }
            break;

        default:
            System.out.println("Invalid choice! Please select 1
            or 2.");
    }

}
}
//Output
MENU:
1.Generate Fibonacci series up to nth term
2.Find product of even digits in a number
  Enter your choice: 1
  Enter the number of terms (n): 10
  Fibonacci Series: 0 1 1 2 3 5 8 13 21 34
```

17. import java.util.*;

```
class Number {
    public static void main() {
        Scanner sc = new Scanner(System.in);

        // Input: Number of elements
        System.out.print("Enter the number of elements (n): ");
        int n = sc.nextInt();
```

```
            int p_Count = 0; // Counter for positive numbers
            int n_Sum = 0;   // Sum of negative numbers

            // Loop to take 'n' numbers as input
            System.out.println("Enter " + n + " numbers:");
            for (int i = 0; i < n; i++) {
                int num = sc.nextInt();
                if (num > 0) {
                    p_Count++; // Count positive numbers
                } else if (num < 0) {
                    n_Sum += num; // Sum negative numbers
                }
            }

            // Print results
          System.out.println("Number of positive numbers: " + p_Count);
            System.out.println("Sum of negative numbers: " + n_Sum);

    }
}
//Output
Enter the number of elements (n): 5
Enter 5 numbers:
-1
2
-3
0
-4
Number of positive numbers: 1
Sum of negative numbers: -8
```

18.
```
import java.util.*;
class chapter9_prog8
 {
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter a number: ");
```

```
        int num = sc.nextInt(),temp,c=0,rem;
        long cube = (long) num * num*num;
        temp = num;
        boolean f=true;
        // Check if the last digits of square match num
        while (temp > 0)
        {
            c++;
            temp /= 10;
        }

        if (num==square%(int)Math.pow(10,c))
        {
            System.out.println(num + " is an Trimorphic Number.");
        }
        else
        {
          System.out.println(num + " is NOT an Trimorphic Number.");
        }

    }
}
//Output
Enter a number: 6
6 is an Automorphic Number.
```

10. Nested Loop

Unsolved Questions

A. 1. c 2. a 3. b 4. c 5. a

B. 1. A loop within a loop is called a **nested loop**.

2. **break** statement is used to terminate a loop.
3. continue statement **skips the remainder** of the current iteration.
4. A **do-while** loop within a do-while loop is known as a nested do-while loop.

C. 1. a. **Nested For vs. Nested Do-While**

A **nested for** loop is simply one for loop placed inside another for loop. The outer loop controls how many times the inner loop is executed in its entirety.

Syntax:

```
for (initialization; condition; update) {
    // Statements for the outer loop

    for (initialization; condition; update) {
        // Statements for the inner loop
    }

    // Possibly more statements in the outer loop
}
```

Nested Do-While Loop: A do-while loop inside another do-while loop.

Each do-while has the form:

```
do {
   // outer loop body
   do {
      // inner loop body
   } while (condition2);
} while (condition1);
```

The inner loop executes completely for each iteration of the outer loop.

b. **Break and continue**

break:

1. Immediately terminates the entire loop (or switch-case).
2. Execution resumes at the statement following the loop/switch.

   Example:

```
for (int i = 0; i < 5; i++) {
    if (i == 3) {
        break; // exits the loop when i == 3
    }

}
```

continue:

1. Skips the remainder of the current iteration and moves on to the next iteration of the loop.

2. The loop itself does not terminate; it just "jumps" to the loop's next iteration check.

   Example:

```
for (int i = 0; i < 5; i++) {
    if (i == 3) {
        continue; // skip printing i == 3
    }
}
```

2. Nested Loop:

   A nested loop is a loop that runs inside the body of another loop.

   Eg.

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 3; j++) {
        // inner loop statements
    }
}
```

3.

```
while (condition1) {
    // statements of the outer loop
    while (condition2) {
        // statements of the inner loop
    }
    // possibly more outer loop statements
}
```

4.

```
class loop
{
    public static void main()
    {
        int i ,j;
        i=1;

        while (i <= 3)// Outer while loop
        {
            j = 1;

            while (j <= 3) // Inner while loop
            {
                if (j == 2)
                {
```

```
                    break;// Break out of the inner loop if j == 2
                }
                System.out.println("j = " + j);
                j++;
            }
            i++;
        }
        System.out.println("i = " + i );

    }
}
//Output
j = 1
j = 1
j = 1
i = 4
```

D. b

E. 1. import java.util.*;

```
class Prime
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int n,i,num,j,c;
        System.out.print("Enter how many numbers you want to input:
        ");
        n = sc.nextInt();
        System.out.println("Enter " + n + " numbers:");
        for (i = 1; i <= n; i++) {
            num=sc.nextInt();
            c=0;
            for(j=1;j<=num;j++)
            {
                if(num%j==0)
                {
                    c++;
                }
```

```
            }
            if(c==2)
            System.out.println(num + " Prime Number ");
            else
            System.out.println(num + " Not Prime Number ");
        }
    }
}
//Output
Enter how many numbers you want to input: 5
Enter 5 numbers:
11
11 Prime Number
14
14 Not Prime Number
17
17 Prime Number
45
45 Not Prime Number
33
33 Not Prime Number
```

2. import java.util.*;

```
class ArmstrongNumber
 {
     public static void main()
     {
        int num,i,sumOfCubes,digit;
        System.out.println("Armstrong numbers between 200 and 1000
        are:");
        for (i = 200; i <= 1000; i++)
        {
            num=i;
            sumOfCubes=0;
            while (num > 0) {
            digit = num % 10;
            sumOfCubes += digit * digit * digit;
            num /= 10;
```

```
            }
                if (sumOfCubes==i) {
                    System.out.print(i + " ");
                }
            }
            System.out.println();
        }
    }
    //Output
    Armstrong numbers between 200 and 1000 are:
    370 371 407
```

3. import java.util.*;

```
    class PalindromeNumber
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);

            int m,n,num,i,rev,digit;
            System.out.println("Enter m and n (m<n)");
            m=sc.nextInt();
            n=sc.nextInt();
            if(m<n)
            {
                System.out.println("Palindrome numbers between "+ m +"
                and "+ n +" are : ");
                for (i = m; i <= n; i++)
                {
                    num=i;
                    rev=0;
                    while (num > 0)
                    {
                        digit = num % 10;
                        rev = rev*10 + digit;
                        num /= 10;
                    }
                    if (rev==i) {
```

```
                    System.out.print(i + " ");
                }
            }
            System.out.println();
        }
        else
        {
            System.out.println("Wrong Range ");
        }
    }
}
//Output
Enter m and n (m<n)
10
50
Palindrome numbers between 10 and 50 are :
11 22 33 44
```

4. import java.util.*;

```
class Niven_Number
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int n,sum=0,temp,r;
        System.out.println("Enter n : ");
        n=sc.nextInt();
        temp=n;
        while (temp > 0)
        {
            r = temp % 10;
            sum += r;
            temp /= 10;
        }
        if (n%sum==0)
            System.out.print(n + " is a Niven Number ");
        else
            System.out.print(n + " is a not Niven Number ");
```

```
    }
}
//Output
Enter n :
111
111 is a Niven Number
```

5. import java.util.*;

```
class Series_Sum
{
    public static void main()
    {
        double sum = 0.0;
        long fact = 1,d;

        for (int i = 1; i <= 10; i++)
        {
            d = 2 * i;
            fact *= (d - 1) * d;
            sum += (double) i / fact;
        }

        System.out.println("Sum of the series: " + sum);
    }
}
//Output
Sum of the series: 0.5876005968219008
```

6. import java.util.*;

```
class SeriesSum
{
    public static void main(int n)
    {
        int  sum = 0,i,j,ind_Sum;
        for (i = 1; i <= n; i++) {
             ind_Sum = 0;
            for (j = 1; j <= (2 * i - 1); j += 2) {
                ind_Sum += j;
            }
```

```
            sum += ind_Sum;
        }
        System.out.println("Sum of the series: " + sum);
    }
}
//Input
n=5
//Output
Sum of the series: 55
```

7. a.
```
class Pattern1
 {
public static void main() {
int i,j,k = 1;
for (i = 1; i <= 4; i++) {
for ( j = 1; j <= i; j++) {
System.out.print(k);
k++;
}
System.out.println();
}
}
}
```

b.
```
class Pattern2
 {
    public static void main() {
        int i,j;
        for (i = 1; i <= 4; i++) {
            for ( j = 4; j >=i; j--) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

c.
```
class Pattern3
 {
public static void main() {
```

```
int i,j,k = 10;
for (i = 4; i >=1; i--) {
for ( j = 1; j <= i; j++) {
System.out.print(k);
k--;
}
System.out.println();
}
}
}
```

d.
```
class Pattern4
 {
    public static void main() {
        int i,j;
        for (i = 1; i <= 4; i++) {
            for ( j = i; j >=1; j--) {
                System.out.print(j);
            }
            System.out.println();
        }
    }
}
```

e.
```
class Pattern5
 {
    public static void main() {
        int i,j;
        for (i = 1; i <= 4; i++) {
            for ( j = 1; j <=i; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

f.
```
class Pattern6
 {
    public static void main() {
```

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

g.
```
class Pattern7
    {
        public static void main() {
            int i,j;
            for (i = 4; i >=1; i--) {
                for ( j = 1; j <=i; j++) {
                    System.out.print("x");
                }
                System.out.println();
            }
        }
    }
```

h.
```
class Pattern2
    {
        public static void main() {
            int i,j,k=65;
            for (i = 1; i <= 4; i++) {
                for ( j = 1; j <=i; j++) {
                    System.out.print((char)k);
                    k++;
                }
                System.out.println();
            }
        }
    }
```

11. Computing and Ethics

Unsolved Questions

A. 1. a 2. b 3. c 4. b 5. b

B. 1. spams and malware 2. licensed 3. Grooming 4. Auto-run
5. Cyberbullying

C. 1. Cyber terrorism refers to the use of the Internet and digital technology to conduct attacks on systems, networks, or data with the intention of causing harm, fear, or disruption, often for political or ideological motives.

2. Software piracy is the unauthorized copying, distribution, or use of software. It is a major concern for software companies as it results in financial losses, reduced revenue, and hinders innovation.

3. Copyright is a legal right that grants creators ownership over their original works, preventing unauthorized use, copying, or distribution.

4. Data privacy refers to the protection of personal or sensitive information from unauthorized access, use, or disclosure.

5. Four Netiquettes are as follows.
   a. Be respectful and polite in online communication.
   b. Do not share personal information online.
   c. Avoid using all caps, as it implies shouting.
   d. Respect copyright laws and do not plagiarize.

6. The Data Protection Act was first implemented in the UK in 1984 and later updated in 1998 and 2018 (GDPR).

7. Four ways to stay safe from malicious code.
   a. Use updated antivirus software.
   b. Do not download files from untrusted sources.
   c. Keep software and operating systems updated.
   d. Avoid clicking on suspicious links or email attachments.

8. Industrial espionage refers to spying or gathering confidential business information from *competitors through illegal or unethical means*.

9. IP snooping is the unauthorized monitoring or tracking of internet activity by intercepting data packets over a network.

10. Cyberwarfare refers to the use of digital attacks by nations or organizations to disrupt or damage the infrastructure, economy, or security of a target country.