Class 10

Answer Key

Touchpad Computer Applications

1. Introduction to Object-Oriented Programming Concepts

MIND DRILL

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

B. 1. Programming 2. Procedural Programming 3. Machine
4. Inheritance 5. Object Oriented

C. 1. Some of the uses of Inheritance in Object-Oriented Programming (OOP)

a. **Code Reusability:** Inheritance allows a new class (child class) to reuse the properties and methods of an existing class (parent class), reducing code duplication.

b. **Avoids Redundancy:** Instead of writing the same code in multiple classes, common functionalities are inherited from a base class.

c. **Polymorphism Support:** It enables method overriding, where a child class provides a specific implementation of a method already defined in the parent class.

2. It is a programming paradigm based on procedures (functions) that operate on data. The program is divided into a series of functions, and the flow is controlled by function calls.

3. Polymorphism in Object-Oriented Programming (OOP) refers to the ability of a function, method, or object to take multiple forms. It allows the same interface to be used for different data types or classes, making code more flexible and scalable.

Real-Life Example: A Person Performing Different Roles.

A single person can act differently in different situations:

a. As a Student in school

b. is a Son/Daughter at home

c. as an Employee at work

4. **Disadvantages of POP**

a. It has global data sharing functions. If any part of the code is changed then it is necessary to make changes in all programs where that function is used.

b. if gives more emphasis on operations rather than data, thus exposing to data to the entire source code, thus reducing data security.

c. It is hard to correlate it with real-world objects.

5. **Difference between POP and OOP**

POP:

1. It divides the program into small parts.
2. It follows top-down approach.
3. It deals with algorithms.
4. It is less secure.
5. **Examples:** C, Fortran

OOP:

1. It divides the program into objects.
2. It follows bottom-up approach.
3. It deals with data.
4. It is more secure
5. Example: C++, Java

D. 1. a. Both A and R are true, and R is the correct explanation of A.

2. d. A is false, but R is true.

2. Elementary Concept of Objects and Classes

MIND DRILL

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

B. 1. A class is a user-defined data type in Object-Oriented Programming (OOP) that serves as a blueprint for creating objects. It defines attributes (data members) and methods (functions) that operate on the data.

2. An object is an instance of a class in Object-Oriented Programming (OOP). It represents a real-world entity with state (attributes) and behavior (methods).

3. The process of creating an instance of a class is called "Instantiation."

C. 1. d. A is false, but R is true.

2. a. Both A and R are true, and R is the correct explanation of A.

3. Values and Data Types

MIND DRILL

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

6. b 7. d 8. c 9. b 10. c

B. 1. Special Characters 2. Different 3. User - Defined

4. Smaller 5. null

C. 1. A String literal in Java is a sequence of characters enclosed in double quotes (" "). It is a predefined object of the String class in Java.A Boolean literal in Java represents one of two values: true or false. It is used for logical expressions and conditions. A Boolean literal in Java represents one of two values: true or false. It is used for logical expressions and conditions.

2. **Declaration:** Declaring a variable by specifying its data type and name without assigning a value. ex: int a;

   Initialization: Assigning a value while declaring a variable. Ex. int a=5;

3. Various type of Tokens in Java are:

   1. **Keywords:** Reserved words in Java with special meaning. For example, class, public, static, if, else, for, return
   2. **Identifiers:** Names for variables, classes, methods, etc. For example, myVariable, Student, calculateSum()
   3. **Literals:** Constant values used in the program.For example, "Hello", 100, 3.14, true, null
   4. **Operators:** Symbols used for operations. For example,.+, -, *, /, ==, &&, `
   5. **Separators (Delimiters):** Special characters used to separate code elements. For example, { }, [ ], ( ), ;, ,
   6. **Comments:** Non-executable text used for explanations.

4. **Primitive Data Types:** These are the basic data types that hold simple values. There are 8 Types of primitive data types. They are byte, short, int, long, float, double, char and Boolean. whereas

   Non-Primitive Data Types: These types store references to objects rather than values. They are Array, class and interface.

5. **Identifier:** Identifiers are the names used for variables, methods, classes, interfaces, etc. Example:// 'Student' is an identifier (class name) and Variable : A variable is a named memory location used to store data. Example: int age = 25; // 'age' is a variable storing value 25.

D. 1. c. A is true, but R is false.

2. b. Both A and R are true, but R is not the correct explanation of A.

4. Operators in Java

MIND DRILL

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

B. 1. ++ % + < 2. (a*a)+(b*b)+(2*a*b) 3. 5 4. true 5. 1

C. 1. a. 32 b. 50

2. 8.0

3. **Difference**

Ternary Operator (?:): An operator that evaluates a condition and returns one of two values. There are three operands (condition ? expr1 : expr2). It acts as a shorthand for an if-else statement. Example: int min = (a < b) ? a : b;

Unary Operator (+, -, ++, --, !): An operator that operates on a single operand (variable or value). One operand (+a, -b, ++c, !d).Used for incrementing, decrementing, negation, or logical NOT operations. Example a++;, b--;, !flag;

4. a. A=(B+C)/2.0*h b. V=22.0/7.0*r*r*h

5. a. relational operator
   b. logical operator
   c. decrement operator
   d. arithmetic operator
   e. logical operator

6. a. 6 b. 2

D. 1. a. Both A and R are true, and R is the correct explanation of A.

2. c. A is true, but R is false.

5. Input in Java

MIND DRILL

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

B. 1. SYNTAX

2. exception

3. logical

4. Scanner nextInt(), nextDouble(), and nextLine()

5. command line

C. 1. Multi-line Comment (/*...*/) : Used for general explanations or temporarily disabling code. Syntax is /* comment */

Documentation Comment (/**...*/):Used to generate documentation using Javadoc. Syntax is /** comment */

2. Scanner sc= new Scanner(System.in);

short s= sc.nextShort();

3. Syntax, Logical and Runtime

4. A logical error (or semantic error) occurs when a program runs without crashing but produces incorrect or unintended results due to flaws in the logic.

ex. print all even numbers up to 10.

```
class a
{
public static void main()
{
      int i;
      for(i=2;i<=10;i++)
      {
          System.out.print(i+" ");
      }
}
}
```

will print : 2 3 4 5 6 7 8 9 10 which is a wrong result.

5. **try:** It is used to enclose code that might throw an exception and always executes first.

catch: It handles the exception thrown in the try block and it executes only if an exception occurs in the try block.

D. 1. d. A is false, but R is true.

2. b. Both A and R are true, but R is not the correct explanation of A.

3. b. Both A and R are true, but R is not the correct explanation of A.

4. a. Both A and R are true, and R is the correct explanation of A.

E. 1. a. Single Line comment

2. b. To set a starting value that can be used in the calculation.

6. Mathematical Library Methods

MIND DRILL

A.

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

B.

1. 5.0
2. 16.0
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. -2.3+(-3.4) = -5.6
10. true

C.

1. In Java, library methods are built-in methods provided by Java's standard libraries (like java.lang, java.util, java.io, etc.) that help perform common operations without writing code from scratch.
2. Math.max(a, b): The Math.max() method returns the greater of two given numbers.Syntax:Math.max(10, 20)

   Math.pow(base, exponent): The Math.pow() method returns the value of the first argument (base) raised to the power of the second argument (exponent).Math.pow(2, 3)
3. 1 (One Argument)
4. A=P*Math.pow((1+r/n),(n*t))
5. a.

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

```
        Scanner sc = new Scanner(System.in);
        double principal,rate,time,comp_mat=0.0;
        // Taking user input
        System.out.print("Enter Principal Amount: ");
        principal= sc.nextDouble();
        System.out.print("Enter Rate of Interest (in % per year):
        ");
        rate = sc.nextDouble();
        System.out.print("Enter Time Period (in years): ");
        time = sc.nextDouble();
        // Calculating Compound Interest Maturity Amount (Compounded
        Annually)
        comp_mat = principal * Math.pow((1 + rate / 100), time);
        // Displaying the results
        System.out.println("Maturity Amount using Compound Interest:
        " + comp_mat);
    }
}
```

b.
```
import java.util.*;
class Cone_Cal
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        double radius,height,slantHeight,surfaceArea,volume;
        // Taking user input
        System.out.print("Enter radius of the cone: ");
        radius = sc.nextDouble();
        System.out.print("Enter height of the cone: ");
        height = sc.nextDouble();
        // Calculating slant height (Pythagoras theorem: l² = r² +
h²)
        slantHeight = Math.sqrt((radius * radius) + (height * height));
```

```
            // Calculating Surface Area: A = πr (r + l)
            surfaceArea = 3.142 * radius * (radius + slantHeight);
            // Calculating Volume: V = (1/3)πr²h
            volume = (1.0 / 3) * 3.142 * radius * radius * height;
            // Displaying the results
            System.out.println("Slant Height: "+ slantHeight);
          System.out.println("Surface Area of the Cone: "+ surfaceArea);
            System.out.println("Volume of the Cone: "+ volume);
        }
    }
```

c.
```
    import java.util.*;
    class number
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            double a,b,c,r;
            // Taking user input
            System.out.print("Enter a, b,c : ");
            a = sc.nextDouble();
            b = sc.nextDouble();
            c = sc.nextDouble();
            r=Math.cbrt(a)+(b*b)-Math.cbrt(c);
            // Displaying the results
            System.out.println(" Result : "+ r);
        }
    }
```

d.
```
    import java.util.*;
    class CircleCalculator
     {
        public static void main()
     {
```

```
        Scanner sc = new Scanner(System.in);
      double diameter, rradius,circum,area;
        // Input diameter
        System.out.print("Enter the diameter of the circle: ");
        diameter = scanner.nextDouble();
        // Calculate radius
        radius = diameter / 2.0;
        // Calculate area and circumference
        circum = 2 * 3.142 * radius;
         area = 3.142 * Math.pow(radius, 2);
        // Display results
        System.out.println("Circumference: " + circum);
        System.out.printf("Area: " +  area);
    }
}
```

e.
```
import java.util.*;
class CircleCalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double diameter, radius, circum, area;
        // Input diameter
        System.out.print("Enter the diameter of the circle: ");
        diameter = sc.nextDouble();
        // Calculate radius
        radius = diameter / 2.0;
        // Calculate area and circumference
        circum = 2 * 3.142 * radius;
        area = 3.142 * Math.pow(radius, 2);
        // Display results
        System.out.println("Circumference: " + circum);
        System.out.println("Area: " + area);
    }
}
```

f.
```
import java.util.*;
class PopulationGrowth
{
    public static void main()
     {
        Scanner sc = new Scanner(System.in);
        double p0, r, t, p;
        // Input values
        System.out.print("Enter the initial population (P0): ");
        p0 = sc.nextDouble();
        System.out.print("Enter the percentage rate of population
        growth (r): ");
       r = sc.nextDouble() / 100.0;  // Convert percentage to decimal
        System.out.print("Enter the time in years (t): ");
        t = sc.nextDouble();
        // Calculate total population
        p = p0 * Math.exp(r * t);  // Using Math.exp() to calculate
        e^(r*t)
        // Display result
        System.out.printf("Total population after :"+ t +" "+ p);
    }
}
```

D. 1. a. Both A and R are true, and R is the correct explanation of A.

2. c. A is true, but R is false.

7. Conditional Constructs in Java

MIND DRILL

A.

1. a	2. b	3. d	4. b	5. c	6. c
7. b	8. a	9. d	10. a	11. c	12. c
13. b	14. b	15. a	16. d		

B. 1. Conditional Operator 2. Switch 3. Conditional Statement
4. False 5. Boolean

C. 1. The difference between if and if-else statements lies in their behavior when the condition is false:

if Statement: Executes a block of code only if the condition is true. If the condition is false, the program simply skips the if block and moves to the next statement.

if-else Statement: Executes one block of code if the condition is true and another block if the condition is false.Ensures that one of the two blocks always executes.

2. A compound statement (also known as a block statement) in Java is a group of multiple statements enclosed within curly braces {}. It allows multiple statements to be treated as a single unit, especially in control structures like if, while, and for.

EX.
```
if (num > 5) {
        System.out.println("Number is greater than 5");
    System.out.println("This is part of the compound statement");
   }
```

3. A fall-through situation occurs in a switch statement when a case does not include a break statement, causing execution to continue to the next case.

example :

```
switch (day) {
    case 1: System.out.println("Monday");
    case 2: System.out.println("Tuesday");   break;
    case 3: System.out.println("Wednesday");    break;
}
```

IF day= 1,

Output will be

Monday

Tuesday

4. The break statement is used in a switch case to prevent fall-through and stop the execution of subsequent cases once a match is found.

EXAMPLE :

```
example :
switch (day) {
    case 1: System.out.println("Monday");break;
```

```
    case 2: System.out.println("Tuesday");   break;
    case 3: System.out.println("Wednesday");    break;
}
```

IF day= 1,

Output will be

Monday

5.
```
import java.util.*;
class check
 {
    public static void main()
    {
         Scanner sc = new Scanner(System.in);
         int n;
        System.out.print("Enter a number: ");
         n = sc.nextInt();
        // Check divisibility by 7 and 8
        if (n % 7 == 0 && n % 8 == 0)
        {
           System.out.println(n + " is divisible by both 7 and 8.");
        }
        else
        {
            System.out.println(n + " is NOT divisible by both 7 and
            8.");
        }
    }
}
```

6.
```
import java.util.*;
class Leap_Year
{
    public static void main()
    {
```

```
        Scanner sc = new Scanner(System.in);
        int year;
        String result;
        // Input: Enter a four-digit year
        System.out.print("Enter a four-digit year: ");
        year= sc.nextInt();
        // Check leap year using the conditional (ternary) operator
        result = (year % 400 == 0) ? "Leap Year" : (year % 4 == 0 &&
        year % 100 != 0) ? "Leap Year" :
            "Not a Leap Year";

        System.out.println(year + " is " + result);
    }
}
```

7.
```
import java.util.*;
class Largest_Smallest
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num1,num2,num3,large, small;
        // Taking input for three numbers
        System.out.print("Enter first number: ");
        num1 = sc.nextInt();
        System.out.print("Enter second number: ");
        num2 = sc.nextInt();
        System.out.print("Enter third number: ");
        num3 = sc.nextInt();
        // Finding the largest number using ternary operator
        large = (num1 > num2 && num1 > num3) ? num1 : (num2 > num3)
? num2 : num3;
        // Finding the smallest number using ternary operator
```

```
        small = (num1 < num2 && num1 < num3) ? num1 : (num2 < num3)
        ? num2 : num3;
        // Displaying results
        System.out.println("Largest number: " + large);
        System.out.println("Smallest number: " + small);
    }
}
```

8.
```
import java.util.*;
class Shopping_Mall
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String c_Name;
        double p_Amount,d_Rate,d_Amount,f_Amount;
        System.out.print("Enter Customer Name: "); // Input Customer
        Name
        c_Name = sc.nextLine();
        System.out.print("Enter Purchase Amount: "); // Input Purchase
        Amount
        p_Amount = sc.nextDouble();
        if (p_Amount <= 2000)  // Determine discount rate based on
        purchase amount
        {
            d_Rate = 2.5;
        }
        else if (p_Amount <= 4000)
        {
            d_Rate = 4.0;
        }
        else if (p_Amount <= 7000)
        {
```

```
            d_Rate = 7.0;
        }
        else
        {
            d_Rate = 10.0;
        }
        d_Amount = (d_Rate / 100) * p_Amount;          // Calculate
        discount amount
        f_Amount = p_Amount - d_Amount;
        System.out.println("\nCustomer Name: " + c_Name); // Output
        results
        System.out.println("Purchase Amount: Rs. " + p_Amount);
        System.out.println("Discount Applied: " + d_Rate + "%");
        System.out.println("Discount Amount: Rs. " + d_Amount);
        System.out.println("Final Amount to Pay: Rs. " + f_Amount);
    }
}
```

9.
```
import java.util.*;
class Telephone_Bill
 {
    public static void main() {
        Scanner sc = new Scanner(System.in);
      String consumer_Name;
      int consumer_Number,number_Calls;
      double rate_Call,billAmount,surcharge,final_Amount;
        // Input Consumer Details
        System.out.print("Enter Consumer Name: ");
        consumer_Name = sc.nextLine();
        System.out.print("Enter Consumer Number: ");
        consumer_Number = sc.nextInt();
        System.out.print("Enter Number of Calls Made: ");
        number_Calls = sc.nextInt();
```

```
        // Determine rate per call based on number of calls
        if (number_Calls > 250) {
            rate_Call = 5.0;
        } else if (number_Calls >= 150) {
            rate_Call = 4.0;
        } else if (number_Calls >= 75) {
            rate_Call = 3.0;
        } else {
            rate_Call = 2.0;
        }
        // Calculate bill amount before surcharge
        billAmount = number_Calls * rate_Call;
        // Calculate surcharge (2.5% of bill amount)
        double surcharge = (2.5 / 100) * billAmount;
        double finalBillAmount = billAmount + surcharge;
        // Display the output
        System.out.println("\n--- Telephone Bill ---");
        System.out.println("Name of Consumer: " + consumerName);
        System.out.println("Consumer Number: " + consumerNumber);
        System.out.println("Number of Calls Made: " + numberOfCalls);
        System.out.println("Bill Amount Before Surcharge: Rs. " +
        billAmount);
        System.out.println("Bill Amount with Surcharge: Rs. " +
        finalBillAmount);
    }
}
```

10.
```
import java.util.*;
class BillPayment
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
```

```
String facility,mode_payment;
double bill_Amount,discount,cashback,final_Amount;
// Input Bill Amount
System.out.print("Enter Bill Amount: ");
 bill_Amount = sc.nextDouble();
// Input Mode of Payment
System.out.print("Enter Mode of Payment (cc/dc/ew/c): ");
mode_payment = sc.next().toLowerCase();
discount = 0.0;
cashback = 0.0;
facility = "No discount";
// Determine facilities based on mode of payment
switch (mode_payment)
{
    case "cc": // Credit Card
        discount = 1.5 / 100 * bill_Amount;
        facility = "1.5% discount";
        break;
    case "dc": // Debit Card
        cashback = 10;
        facility = "Rs. 10 cashback";
        break;
    case "ew": // E-Wallet
        cashback = 20;
        facility = "Rs. 20 cashback";
        break;
    case "c": // Cash
        facility = "No discount";
        break;
    default:
        System.out.println("Invalid payment mode! Please
        enter cc, dc, ew, or c.");
```

```
                return;
            }
           // Calculate final amount after applying discount or cashback
            final_Amount = bill_Amount - discount - cashback;
            // Display the output
            System.out.println("\n--- Bill Payment Details ---");
            System.out.println("Bill Amount: Rs. " + bill_Amount);
            System.out.println("Mode of Payment: " + mode_payment.
            toUpperCase());
            System.out.println("Facility Provided: " + facility);
            System.out.println("Discount Applied: Rs. " + discount);
            System.out.println("Cashback Received: Rs. " + cashback);
            System.out.println("Final Amount to be Paid: Rs. " + final_
            Amount);
        }
    }
```

11.
```
    import java.util.*;
    class Electricity_Bill
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            int prev_Read,cur_Read,units_Cons;
            double bill_Amt=0.0;
            // Input Previous and Current Meter Readings
           System.out.print("Enter Previous Month's Meter Reading: ");
            prev_Read = sc.nextInt();
            System.out.print("Enter Current Month's Meter Reading: ");
            cur_Read= sc.nextInt();
            // Calculate units consumed
            units_Cons = cur_Read - prev_Read;
            if (units_Cons < 0)
```

```
        {
            System.out.println("Invalid readings! Current reading
            should be greater than or equal to previous reading.");
            return;
        }
        // Calculate bill based on unit slabs
        if (units_Cons <= 100) {
            bill_Amt = units_Cons * 2.0;
        } else if (units_Cons <= 250) {
            bill_Amt = (100 * 2.0) + ((units_Cons - 100) * 3.5);
        } else if (units_Cons <= 650) {
            bill_Amt = (100 * 2.0) + (150 * 3.5) + ((units_Cons -
            250) * 5.0);
        } else {
            bill_Amt = (100 * 2.0) + (150 * 3.5) + (400 * 5.0) +
            ((units_Cons - 650) * 7.5);
        }
        // Display the output
        System.out.println("\n--- Electricity Bill ---");
        System.out.println("Previous Meter Reading: " + prev_Read);
        System.out.println("Current Meter Reading: " + cur_Read);
        System.out.println("Units Consumed: " + units_Cons);
        System.out.println("Total Bill Amount: Rs. " + bill_Amt);
    }
}
```

12.
```
import java.util.*;
class SalesC
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int n_Sales;
        double s_Amt,c_Rate=0.0,c_Amt;
```

```
        String gift;
        // Input number of products sold
        System.out.print("Enter the number of products sold: ");
        n_Sales = sc.nextInt();
        // Input total selling amount
        System.out.print("Enter the total selling amount: ");
        s_Amt = sc.nextDouble();
        gift = "No Gift";
        // Determine commission and gift based on number of sales
        if (n_Sales <= 50) {
            c_Rate = 5.0;
            gift = "A Parker Pen";
        } else if (n_Sales <= 75) {
            c_Rate = 7.5;
            gift = "A Micro SD Card";
        } else if (n_Sales <= 100) {
            c_Rate = 10.0;
            gift = "A Mobile";
        } else {
            c_Rate = 15.0;
            gift = "A Laptop";
        }
        // Calculate commission amount
        c_Amt = (c_Rate / 100) * s_Amt;
        // Display the results
        System.out.println("\n--- Sales Commission Details ---");
        System.out.println("Number of Products Sold: " + n_Sales);
        System.out.println("Total Selling Amount: Rs. " + s_Amt);
        System.out.println("Commission Rate: " + c_Rate + "%");
        System.out.println("Commission Earned: Rs. " + c_Amt);
        System.out.println("Gift Received: " + gift);
    }
}
```

13.
```
import java.util.*;
class Volume_Cal
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch;
        double r,h,volS,volC;
        do {
            // Display Menu
            System.out.println("\n--- Volume Calculator ---");
            System.out.println("1. Volume of Sphere");
            System.out.println("2. Volume of Cylinder");
            System.out.println("3. Volume of Cone");
            System.out.println("4. Exit");
            System.out.print("Enter your choice (1-4): ");
            ch = sc.nextInt();
            switch (ch) {
                case 1:
                    // Volume of Sphere: V = (4/3)πr³
                    System.out.print("Enter radius of the sphere:
                    ");
                    r = sc.nextDouble();
                    volS = (4.0 / 3.0) * Math.PI * Math.pow(r, 3);
                   System.out.println("Volume of Sphere: " + volS);
                    break;
                case 2:
                    // Volume of Cylinder: V = πr²h
                    System.out.print("Enter radius of the cylinder:
                    ");
                    r = sc.nextDouble();
                    System.out.print("Enter height of the cylinder:
                    ");
```

```
                    h = sc.nextDouble();
                    volC = Math.PI * Math.pow(r, 2) * h;
                 System.out.println("Volume of Cylinder: " + volC);
                    break;
                case 3:
                    // Volume of Cone: V = (πr²h) / 3
                   System.out.print("Enter radius of the cone: ");
                    r = sc.nextDouble();
                   System.out.print("Enter height of the cone: ");
                    h = sc.nextDouble();
                    volC = (Math.PI * Math.pow(r, 2) * h) / 3;
                    System.out.println("Volume of Cone: " + volC);
                    break;
                case 4:
                    System.out.println("Exiting the program. Thank
                    you!");
                    break;
                default:
                    System.out.println("Invalid choice! Please enter
                    a number between 1 and 4.");
            }
        } while (ch != 4);
    }
}
```

14.
```
import java.util.*;
class Electronics_Shop
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String name, address;
        double p_Amt,d_Rate=0.0;
```

```
        char type;
        // Input customer details
        System.out.print("Enter Customer Name: ");
        name = sc.nextLine();
        System.out.print("Enter Customer Address: ");
        address = sc.nextLine();
        System.out.print("Enter Purchase Amount: ");
        p_Amt = sc.nextDouble();
         System.out.print("Enter Type of Purchase (L for Laptop, D
for Desktop): ");
        type = sc.next().charAt(0);
        // Determine discount based on purchase amount
        if (p_Amt >= 0 && p_Amt <= 25000) {
            d_Rate = (type == 'L' || type == 'l') ? 0.0 : 5.0;
        } else if (p_Amt >= 25001 && p_Amt <= 57000) {
            d_Rate = (type == 'L' || type == 'l') ? 5.0 : 7.6;
        } else if (p_Amt >= 57001 && p_Amt <= 100000) {
            d_Rate = (type == 'L' || type == 'l') ? 7.5 : 10.0;
        } else if (p_Amt > 100000) {
            d_Rate = (type == 'L' || type == 'l') ? 10.0 : 15.0;
        } else {
            System.out.println("Invalid purchase amount!");
            return;
        }
        // Calculate discount and net amount
        double discount = (d_Rate / 100) * p_Amt;
        double netAmount = p_Amt - discount;
        // Display results
        System.out.println("\n--- Bill Details ---");
        System.out.println("Customer Name: " + name);
        System.out.println("Customer Address: " + address);
        System.out.println("Purchase Type: " + (type == 'L' || type
```

```
            == 'l' ? "Laptop" : "Desktop PC"));
            System.out.println("Purchase Amount: Rs. " + p_Amt);
            System.out.println("Discount Applied: " + d_Rate + "%");
            System.out.println("Discount Amount: Rs. " + discount);
            System.out.println("Net Amount to be Paid: Rs. " + netAmount);
        }
    }
```

15.
```
import java.util.*;
class Temperature_Con
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch;
        double fah,cel;
        while (true) {
            // Display menu
            System.out.println("\n--- Temperature Converter ---");
            System.out.println("1. Convert Fahrenheit to Celsius");
            System.out.println("2. Convert Celsius to Fahrenheit");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            ch = sc.nextInt();
            switch (ch) {
                case 1:
                    System.out.print("Enter temperature in Fahrenheit:
                    ");
                    fah= sc.nextDouble();
                    cel = (fah - 32) * 5 / 9;
                    System.out.println("Temperature in Celsius: "+
                    cel);
                    break;
```

```
                    case 2:
                        System.out.print("Enter temperature in Celsius:
                        ");
                        cel = sc.nextDouble();
                        fah = (cel * 9 / 5) + 32;
                        System.out.printf("Temperature in Fahrenheit: "+
                        fah);
                        break;
                    case 3:
                        System.out.println("Exiting the program. Thank
                        you!");
                        System.exit(0);
                    default:
                        System.out.println("Invalid choice! Please enter
                        a valid option.");
                }
            }
        }
    }
```

16.
```
import java.util.*;
class Number_Checker {
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int ch, num1,num2;
        while (true) {
            System.out.println("\n--- Number Checker ---");
            System.out.println("1. Check if a number is a Palindrome");
            System.out.println("2. Check if a number is a Perfect
            number");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            ch = sc.nextInt();
            switch (ch) {
```

```
            case 1:
                System.out.print("Enter a number to check for
                Palindrome: ");
                num1 = sc.nextInt();
                if (isPalindrome(num1)) {
                    System.out.println(num1 + " is a Palindrome
                    number.");
                } else {
                    System.out.println(num1 + " is NOT a Palindrome
                    number.");
                }
                break;
            case 2:
                System.out.print("Enter a number to check for
                Perfect number: ");
                num2 = sc.nextInt();
                if (isPerfect(num2)) {
                    System.out.println(num2 + " is a Perfect
                    number.");
                } else {
                    System.out.println(num2 + " is NOT a Perfect
                    number.");
                }
                break;
            case 3:
                System.out.println("Exiting the program. Thank
                you!");
                sc.close();
                return;
            default:
                System.out.println("Invalid choice! Please enter
                a valid option.");
        }
```

```
        }
    }
    public static boolean isPalindrome(int num) {
        int original = num, rev = 0;
        while (num > 0) {
            int digit = num % 10;
            rev = rev * 10 + digit;
            num /= 10;
        }
        return original == rev;
    }
    public static boolean isPerfect(int num) {
        int sum = 0;
        for (int i = 1; i < num; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }
        return sum == num;
    }
}
```

17.
```
import java.util.*;
class Buzz_GCD
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int ch,num1,num2,a,b,gcd;
        while (true) {
            System.out.println("\n--- Number Operations ---");
            System.out.println("1. Check if a number is a BUZZ
            number");
```

```
System.out.println("2. Calculate GCD of two numbers");
System.out.println("3. Exit");
System.out.print("Enter your choice: ");
ch = sc.nextInt();
switch (ch) {
    case 1:
        System.out.print("Enter a number to check for
        BUZZ number: ");
        num1 = sc.nextInt();
        if ((num1 % 10 == 7) || (num1 % 7 == 0)) {
        System.out.println(num1 + " is a BUZZ number.");
        } else {
            System.out.println(num1 + " is NOT a BUZZ
            number.");
        }
        break;
    case 2:
        System.out.print("Enter first number: ");
        a = sc.nextInt();
        System.out.print("Enter second number: ");
        b = sc.nextInt();
        System.out.print("The GCD of " + a + " and " +
        b + " is: ");
        while (b != 0)
        {
            int temp = b;
            b = a % b;
            a = temp;
        }
        gcd = a;
        System.out.println( gcd);
        break;
```

```
                    case 3:
                        System.out.println("Exiting the program. Thank
                        you!");
                        return;
                    default:
                        System.out.println("Invalid choice! Please enter
                        a valid option.");
                }
            }
        }
    }
```

D. 1. a. Both A and R are true, and R is the correct explanation of A.

2. c. A is true, but R is false.

8. Iterative Constructs in Java

MIND DRILL

A.

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

B. 1. 10 2. Condition 3. Iteration 4. Outside 5. 4

C. 1. a.

```
class NumberSeries
 {
    public static void main()
    {
        int i;
        for ( i = 1; i <= 10; i++)
        {
            System.out.print(i + " ");
        }
    }
}
```

b.

```
lass NumberSeries
```

```
    {
        public static void main()
        {
            int i;
             for (i = 1; i <= 19; i += 2)
             {
                System.out.print(i + " ");
            }
        }
    }
```

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

d.
```
class NumberSeries
    {
        public static void main()
        {
            int i;
             for (i = 5; i <= 50; i += 5)
             {
                System.out.print(i + " ");
            }
        }
    }
```

e.
```
class FractionSeries
    {
```

```
    public static void main()
    {
        int i;
        for (i = 1; i <= 10; i++)
        {
            System.out.print(i + "/" + (i*3));
            if (i < 10)
            {
                System.out.print(", ");
            }
        }
    }
}
```

f.
```
class FractionSeries
 {
    public static void main()
    {
        int i;
        for (i = 2; i <= 10; i=i+2)
        {
            System.out.print(i + "/" + (i+2));
            if (i < 10)
            {
                System.out.print(", ");
            }
        }
    }
}
```

2. a.
```
class Series_Sum
{
    public static void main()
    {
        int i,s=0;
        for (i = 1; i <= 20; i++)
```

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

b.
```
class Series_product
    {
        public static void main()
        {
            int i,p=1;
            for (i = 1; i <= 9; i=i+2)
            {
                p=p*i;
            }
            System.out.print("p = " + p);
        }
    }
```

c.
```
class Series_product
    {
        public static void main()
        {
            int i,t=0,s=0;
            for (i = 1; i <= 5; i=i+1)
            {
             t=t*10+1;
                s=s+t;
            }
            System.out.print("s = " + s);
        }
    }
```

d.
```
class product
    {
     public static void main()
```

```
    {
        int i,p=1;
        for (i = 1; i <= 5; i=i+1)
        {
         p=p*(int)Math.pow(2,i);
        }
        System.out.print("p = " + p);
    }
}
```

e.
```
class Series_product
{
    public static void main()
    {
        int i,t=1,s=0;
        for (i = 1; i <= 5; i=i+1)
        {
            t=t*i;
            s=s+t;
        }
        System.out.print("s = " + s);
    }
}
```

f.
```
class Series_product
{
    public static void main()
    {
        double i,t1=1,t2=0,s=0;
        for (i = 1; i <= 5; i=i+1)
        {
            t1=t1*i;
            t2=t2+i;
            s=s+(t2/t1);
        }
        System.out.print("s = " + s);
```

```
    }
}
```

3.
```
import java.util.*;
class LCM
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num1, num2,lcm,max;
        // Taking input from user
        System.out.print("Enter first number: ");
        num1 = sc.nextInt();
        System.out.print("Enter second number: ");
        num2 = sc.nextInt();
        // Find max of two numbers
        max = Math.max(num1, num2);
        lcm = max;
       // Keep increasing lcm until it is divisible by both numbers
        while (lcm % num1 != 0 || lcm % num2 != 0)
        {
            lcm++;
        }
        // Display result
        System.out.println("LCM of " + num1 + " and " + num2 + " is:
        " + lcm);
    }
}
```

4.
```
import java.util.*;
class ReverseDigits
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int n,digit;
```

```
        boolean first=true;
        System.out.print("Enter a number: ");
        n = sc.nextInt();
        System.out.print("Output: ");
        while (n > 0) {
            digit = n % 10; // Get last digit
            if (first) {
                System.out.print(digit); // for the last digit to be
                printed in the first
                first = false;
            } else {
                System.out.print("," + digit);
            }
            n /= 10; // Remove last digit
        }
        System.out.print(",");
    }
}
```

5.
```
import java.util.*;
class SumProduct
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int n,even=0,odd=1,digit;
        boolean has=false;
        System.out.print("Enter a number: ");
        n = sc.nextInt();
        while (n > 0)
        {
            digit = n % 10; // Extract last digit
            if (digit % 2 == 0)
            {
                even+= digit; // Sum even digits
```

```
            } else {
                odd *= digit; // Multiply odd digits
                has = true;
            }
            n /= 10; // Remove last digit
        }
        System.out.println("Sum of the even digits is: " + even);
        System.out.println("Product of the odd digits is: " + (has
        ? odd : 0));
    }
}
```

6. a.
```
class calculation
{
    public static void main(int a, int n)
    {
        // Calculating sum of series
        int sum = 0,i,term=1;
        for (i = 1; i <= n; i++)
        {
            term *= a; // Compute a^i iteratively
            sum += term; // Add to sum
        }
        // Output the result
        System.out.println("Sum of the series S = " + sum);
    }
}
```

b.
```
class calculation
{
    public static void main(int x, int n)
    {
        // Calculating sum of series
        int sum = 0,i;
        for (i = 1; i <= n; i++)
        {
```

```
            sum += i*(x*x); // Add to sum
        }
        // Output the result
        System.out.println("Sum of the series S = " + sum);
    }
}
```

c.
```
class calculation
{
    public static void main(int x, int n)
    {
        // Calculating sum of series
        int p = 1,i;
        for (i = 1; i <= n; i++)
        {
            p *= (int)Math.pow(i,i); // Add to sum
        }
        // Output the result
        System.out.println("Product of the series = " + sum);
    }
}
```

d.
```
class calculation
{
    public static void main()
    {
        // Calculating sum of series
        double sum = 0,i;
        for (i = 2; i <= 10; i=i+2)
        {
            sum =sum + 1/Math.sqrt(i); // Add to sum
        }
        // Output the result
        System.out.println("Sum of the series S = " + sum);
    }
}
```

e.
```
class calculation
{
    public static void main()
    {
        // Calculating sum of series
        double sum = 0,i,f,j;
        for (i = 2; i <= 10; i=i+2)
        {
            f=1;
            for(j=1;j<=i;j++)
            {
                f=f*j;
            }
            sum =sum + 1/f; // Add to sum
        }
        // Output the result
        System.out.println("Sum of the series S = " + sum);
    }
}
```

f.
```
class calculation
{
    public static void main(double a)
    {
        // Calculating sum of series
        double p = 1,i;
        for (i = 2; i <= 10; i=i+2)
        {
            p = p * a/Math.sqrt(i); // Add to sum
        }
        // Output the result
        System.out.println("Product of the series  = " + p);
    }
}
```

g.
```
class calculation1
{
    public static void main(int a, int n)
    {
        // Calculating sum of series
        int sum = 0,i,term=1;
        for (i = 1; i <= n; i++)
        {
            term *= a; // Compute a^i iteratively
            if(i%2==0)
                sum -= term; // Add to sum
            else
                sum +=term;
        }
        // Output the result
        System.out.println("Sum of the series S = " + sum);
    }
}
```

h.
```
class calculation
{
    public static void main(int x, int n)
    {
        // Calculating sum of series
        int p = 1,i;
        for (i = 1; i <= n; i=i+2)
        {
            sum += (x+i); // Add to sum
        }
        // Output the result
        System.out.println("Sum of the series S = " + sum);
    }
}
```

7.
```
import java.util.*;
class Niven
```

```
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num,sum_digits=0,temp;
        // Taking input from user
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        temp = num;
        // Calculate sum of digits
        while (temp > 0)
        {
            sum_digits += temp % 10; // Extract last digit and add
            to sum
            temp /= 10; // Remove last digit
        }
       // Check if the number is divisible by the sum of its digits
        if (num % sum_digits == 0) {
            System.out.println(num + " is a Niven number.");
        } else {
            System.out.println(num + " is not a Niven number.");
        }
    }
}
```

8.
```
import java.util.*;
class Automorphic
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num,square,t,c=0;
        String squareStr;
        // Taking input from user
        System.out.print("Enter a number: ");
```

```
            num = sc.nextInt();
            square = num * num;
            // Convert numbers to strings for easy comparison
            t=num;
            while(t>0)
            {
                c++;
                t=t/10;
            }
            // Check if square ends with the original number
            if (num== (int)square%Math.pow(10,c))
            {
                System.out.println(num + " is an Automorphic number.");
            }
            else
            {
                System.out.println(num + " is not an Automorphic number.");
            }
        }
    }
```

9.
```
import java.util.*;
class krishnamurthy
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Taking input from user
        int num,originalNum,sum = 0,digit,i,f;
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        originalNum = num;
        // Compute sum of factorial of digits
        while (num > 0)
        {
```

```
            digit = num % 10;
            i=1;
            f=1;
            while(i<=digit)
            {
                f=f*i;
                i++;
            }
            sum += f;
            num /= 10;
        }
        // Check if sum of factorials equals the original number
        if (sum == originalNum) {
            System.out.println(originalNum + " is a Krishnamurthy
            Number.");
        } else {
            System.out.println(originalNum + " is not a Krishnamurthy
            Number.");
        }
    }
}
```

10.
```
import java.util.*;
class SpyNumber
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num,sum=0,pro=1,o_num,digit;
        // Taking input from user
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        o_num = num;
        // Compute sum and product of digits
        while (num > 0)
```

```
        {
            digit = num % 10;
            sum += digit;
            pro *= digit;
            num /= 10;
        }
        // Check if sum and product are equal
        if (sum == pro) {
            System.out.println(o_num + " is a Spy Number.");
        } else {
            System.out.println(o_num + " is not a Spy Number.");
        }
    }
}
```

11.
```
import java.util.*;
class MenuDriven
 {
    // Function to check if a number is Perfect
    public static boolean isPerfect(int num) {
        int sum = 0;
        for (int i = 1; i < num; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }
        return sum == num;
    }
    // Function to check if a number is Trimorphic
    public static boolean isTrimorphic(int num)
    {
        int cube = num * num * num;
        int i=num,c=0;
        while(i>0)
        {
```

```
            c++;
            i=i/10;
        }
        return (cube % Math.pow(10, c)== num);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Menu display
        System.out.println("Choose an option:");
        System.out.println("1. Check Perfect Number");
        System.out.println("2. Check Trimorphic Number");
        System.out.print("Enter your choice (1 or 2): ");
        int choice = scanner.nextInt();
        // Input number from user
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        // Perform action based on user choice
        switch (choice) {
            case 1:
                if (isPerfect(num)) {
                    System.out.println(num + " is a Perfect Number.");
                } else {
                    System.out.println(num + " is not a Perfect
                    Number.");
                }
                break;
            case 2:
                if (isTrimorphic(num)) {
                    System.out.println(num + " is a Trimorphic
                    Number.");
                } else {
                    System.out.println(num + " is not a Trimorphic
                    Number.");
                }
```

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

12.
```
import java.util.*;
class MenuDrivenSeries
 {
    // Method to compute sum of the series: 1 + 12 + 123 + 1234 +
    ...
    int sumSeries(int n) {
        int total = 0, num = 0,i;
        for (i = 1; i <= n; i++) {
            num = num * 10 + i;  // Constructing numbers like 1, 12,
            123, etc.
            total += num;
        }
        return total;
    }
    // Method to compute product of the series: (1/1) * (2/4) * (3/9)
    * ...
    double productSeries(int n) {
        double product = 1.0;
        int i;
        for (i = 1; i <= n; i++) {
            product *= (double) i / (i * i);  // (i / i^2)
        }
        return product;
    }
    void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (true) {
```

```
                System.out.println("\nMenu:");
                System.out.println("1. Compute sum of series: 1 + 12 +
                123 + ... + nth term");
                System.out.println("2. Compute product of series: (1/1)
                * (2/4) * (3/9) * ... nth term");
                System.out.println("3. Exit");
                System.out.print("Enter your choice: ");
                int ch = sc.nextInt();
                switch (ch) {
                    case 1:
                        System.out.print("Enter the number of terms: ");
                        int n1 = sc.nextInt();
                        System.out.println("Sum of the series: " +
                        sumSeries(n1));
                        break;
                    case 2:
                        System.out.print("Enter the number of terms: ");
                        int n2 = sc.nextInt();
                        System.out.println("Product of the series: " +
                        productSeries(n2));
                        break;
                    case 3:
                        System.out.println("Exiting program.");
                        sc.close();
                        return;
                    default:
                        System.out.println("Invalid choice. Please try
                        again.");
                }
            }
        }
    }
```

13.
```
import java.util.*;
public class SeriesShow
{
```

```
public static void main() {
    Scanner sc = new Scanner(System.in);
    while (true) {
        System.out.println("\nMenu:");
        System.out.println("1. Print first N natural numbers with
        their squares and cubes");
        System.out.println("2. Compute sum of series: S = 1/1^3
        - 1/2^3 + 1/3^3 - ... + 1/n^3");
        System.out.println("3. Exit");
        System.out.print("Enter your choice: ");
        int choice = sc.nextInt();
        switch (choice) {
            case 1:
               System.out.print("Enter the number of terms: ");
                int n1 = sc.nextInt();
                for (int i = 1; i <= n1; i++) {
                    System.out.println(i + "\t" + (i * i) + "\t"
                    + (i * i * i));
                }
                break;
            case 2:
                System.out.print("Enter the number of terms: ");
                int n2 = sc.nextInt();
                double sum = 0.0;
                for (int i = 1; i <= n2; i++) {
                    if(i%2==1)
                        sum += 1 / Math.pow(i, 3); // Alternating
                        sign formula
                    else
                        sum -= 1 / Math.pow(i, 3);
                }
                System.out.println("Sum of the series: " + sum);
                break;
```

```
                    case 3:
                        System.out.println("Exiting program.");
                        System.exit(0);
                    default:
                        System.out.println("Invalid choice. Please try
                        again.");
                }
            }
        }
    }
```

14.
```
import java.util.*;
class SeriesSum {
    // Method to compute factorial of a number
    static long factorial(int num) {
        long fact = 1,i;
        for (i = 1; i <= num; i++) {
            fact *= i;
        }
        return fact;
    }
    // Method to compute the sum of the series
    double computeSeriesSum(double x, int n) {
        double sum = 0.0;
        int i;
       for (i = 2; i <= n; i++) { // Start from i = 2 as per formula
            sum += (x + Math.pow(x, i)) / factorial(i);
        }
        return sum;
    }
    void main() {
        Scanner sc = new Scanner(System.in);
        double x,result;
        int n;
        // Taking input
```

```
            System.out.print("Enter the value of x: ");
            x = sc.nextDouble();
            System.out.print("Enter the value of N: ");
            n = sc.nextInt();
            // Computing and displaying result
            result = computeSeriesSum(x, n);
            System.out.println("Sum of the series: " + result);
        }
    }
```

15.
```
import java.util.*;
class Employee
 {
    // Method to calculate tax based on salary
    public static double calculateTax(double salary)
    {
        double tax = 0.0;
        if (salary > 65000)
         {
            tax += (salary - 65000) * 0.10;  // 10% on amount above
            65000
            salary = 65000;
        }
        if (salary > 35000)
         {
            tax += (salary - 35000) * 0.075;  // 7.5% on amount above
            35000
            salary = 35000;
        }
        if (salary > 20000)
         {
            tax += (salary - 20000) * 0.055;  // 5.5% on amount above
            20000
        }
        return tax;
```

```
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        double salaries;
        // Input salaries for 5 employees
        for (int i = 0; i < 5; i++)
        {
            System.out.print("Enter salary of employee " + (i + 1)
            + ": ");
            salaries= sc.nextDouble();
            double tax = calculateTax(salaries);
            System.out.println((i + 1)+" : "+ " : " + salaries +" :
            " + tax);
        }
    }
}
```

9. Nested Loop

MIND DRILL

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

B. 1.

```
import java.util.*;
class NeonNumber
{
    public static boolean isNeon(int num)
    {
        int square,sum=0;
        square = num * num;
        while (square > 0)
        {
            sum += square % 10;
            square /= 10;
```

```
        }
        return sum == num;
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num,i;
        System.out.println("Enter 10 numbers:");
        for (i = 0; i < 10; i++) {
            System.out.print("Enter number :" + (i+1) +" : ");
            num = sc.nextInt();
            if (isNeon(num)) {
                System.out.println(num + " is a Neon Number.");
            }
        }
    }
}
```

2.
```
import java.util.*;
class MagicNumber
 {
    public static int sumOfDigits(int num)
    {
        int sum = 0;
        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }
        return sum;
    }
    public static boolean isMagic(int num) {
        while (num > 9) {
            num = sumOfDigits(num);
        }
        return num == 1;
    }
```

```
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int num;
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        if (isMagic(num)) {
            System.out.println(num + " is a Magic Number.");
        } else {
            System.out.println(num + " is NOT a Magic Number.");
        }
    }
}
```

3.
```
import java.util.*;
class PalindromeNumbers
{
    public static boolean isPalindrome(int num)
    {
        int original = num, reverse = 0;
        while (num > 0)
        {
            reverse = reverse * 10 + num % 10;
            num /= 10;
        }
        return original == reverse;
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int m,n,i;
        System.out.print("Enter range (m and n): ");
        m = sc.nextInt();
        n = sc.nextInt();
        System.out.println("Palindromic numbers between " + m + "
        and " + n + ":");
```

```
            for (i = m; i <= n; i++) {
                if (isPalindrome(i)) {
                    System.out.print(i + " ");
                }
            }
        }
    }
```

4.
```
import java.util.*;
class ArmstrongNumbers {
    public static boolean isArmstrong(int num) {
        int sum = 0, temp = num, digits , nodigits;
        String n = String.valueOf(num);
        nodigits=n.length();
        while (temp > 0) {
            sum += Math.pow(temp % 10, nodigits);
            temp /= 10;
        }
        return sum == num;
    }
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int n,sum=0,num,i;
        System.out.print("Enter how many numbers: ");
        n = sc.nextInt();
        System.out.println("Enter " + n + " numbers:");
        for (i = 0; i < n; i++) {
            num = sc.nextInt();
            if (isArmstrong(num)) {
                sum += num;
            }
        }
        System.out.println("Sum of Armstrong numbers: " + sum);
    }
}
```

5.
```
import java.util.*;
class TwinPrime {
    public static boolean isPrime(int num)
    {
        if (num < 2)
        return false;
        for (int i = 2; i * i <= num; i++)
        {
            if (num % i == 0)
            return false;
        }
        return true;
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num1, num2;
        System.out.print("Enter two numbers: ");
        num1 = sc.nextInt();
        num2 = sc.nextInt();
        if (Math.abs(num1 - num2) == 2 && isPrime(num1) && isPrime(num2))
        {
            System.out.println(num1 + " and " + num2 + " are Twin
            Primes.");
        } else {
            System.out.println(num1 + " and " + num2 + " are NOT Twin
            Primes.");
        }
    }
}
```

6. 1.
```
class Pattern_a
{
    public static void main()
    {
```

```
            for (int i = 1; i <= 4; i++)
            {
                for (int j = 1; j <= i; j++)
                {
                    System.out.print(i * j + " ");
                }
                System.out.println();
            }
        }
    }
```

2.
```
class Pattern_b
    {
        public static void main()
        {
            for (int i = 1; i <= 5; i++)
            {
                for (int j = i; j <= 5; j++)
                {
                    System.out.print(j);
                }
                System.out.println();
            }
        }
    }
```

3.
```
class Pattern_c
    {
        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();
        }
    }
}
```

4.
```
public class Pattern_d
 {
    public static void main()
    {
        int i,j,n;
        for (i = 1, n = 5; i <= 5; i++, n--)
        {
            for ( j = 1; j <= n; j++)
            {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

5.
```
class Pattern_e
 {
    public static void main()
    {
        int i,j;
        for (i = 1; i <= 5; i++)
        {
            for ( j = 1; j <= i; j++)
            {
                System.out.print((i + j) % 2);
            }
            System.out.println();
        }
    }
}
```

6.
```
class Pattern_f
{
    public static void main()
    {
        int i,j;
        for (i = 5; i >= 1; i--)
        {
            for (j = 1; j <= i; j++)
            {
                if (j % 2 == 0)
                    System.out.print("#");
                else
                    System.out.print("x");
            }
            System.out.println();
        }
    }
}
```

7.
```
class Pattern_g
{
    public static void main()
    {
        char ch = 'A';
        int i,j;
        for (i = 1; i<= 4; i++)
        {
            for (j = 1; j <= i; j++)
            {
                System.out.print(ch++);
            }
            System.out.println();
        }
    }
}
```

C. 1. a. Both A and R are true, and R is the correct explanation of A.
2. a. Both A and R are true, and R is the correct explanation of A.

10. User-defined Methods

MIND DRILL

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

B. 1. Method 2. Access Specifier 3. Built-In 4. Return 5. Primitive

C. 1. c. A is true, but R is false.
2. a. Both A and R are true, and R is the correct explanation of A.
3. d. A is false, but R is true.
4. a. Both A and R are true, and R is the correct explanation of A.
5. d. A is false, but R is true.

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

Unsolved Program

1.
```
import java.util.*;
class AutomorphicNumber
 {
    public static void automorphic(int num)
    {
         String numStr,squareStr;
         int square;
square = num * num;
         numStr = String.valueOf(num);
         squareStr = String.valueOf(square);
         if (squareStr.endsWith(numStr))
         {
             System.out.println(num + " is an Automorphic number.");
         }
         else
```

```
            {
                System.out.println(num + " is not an Automorphic number.");
            }
        }
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            int num;
            System.out.print("Enter a number: ");
            num = sc.nextInt();
            automorphic(num);
        }
    }
```

2.
```
    import java.util.*;
    class Calculator
    {
        public static int sum(int a, int b)
        {
            return a + b;
        }
        public static int subtract(int a, int b)
        {
            return a - b;
        }
        public static int product(int a, int b)
        {
            return a * b;
        }
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            int a,b,ch;
```

```
            System.out.print("Enter two numbers: ");
            a = sc.nextInt();
            b = sc.nextInt();
            System.out.println("Choose operation: 1. Sum 2. Subtract 3.
            Product");
            ch = sc.nextInt();
            switch (ch)
            {
               case 1 : System.out.println("Sum: " + sum(a, b)); break;
                case 2 : System.out.println("Difference: " + subtract(a,
                b)); break;
                case 3 : System.out.println("Product: " + product(a, b));
                break;
                default : System.out.println("Invalid choice");
            }
        }
}
// OTHER METHOD
import java.util.*;
class Calculator
{
    public static int sum(int a, int b)
    {
        return a + b;
    }
    public static int subtract(int a, int b)
    {
        return a - b;
    }
    public static int product(int a, int b)
    {
        return a * b;
```

```
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int a,b,ch;
        System.out.print("Enter two numbers: ");
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println("Choose operation: 1. Sum 2. Subtract 3.
        Product");
        ch = sc.nextInt();
        switch (ch)
        {
            case 1 -> System.out.println("Sum: " + sum(a, b));
            case 2 -> System.out.println("Difference: " + subtract(a,
            b));
            case 3 -> System.out.println("Product: " + product(a,
            b));
            default -> System.out.println("Invalid choice");
        }
    }
}
```

[NOTE: IS → IS USED, THEN BREAK IS NOT NEEDED IN SWITCH CASE]

3.
```
import java.util.*;
class VowelConsonant
{
    public static void vowel(String str)
    {
        System.out.print("Vowels: ");
        for (int i=0;i<str.length();i++)
        {
            if ("AEIOUaeiou".indexOf(str.charAt(i)) != -1)
```

```
            {
                System.out.print(str.charAt(i) + " ");
            }
        }
        System.out.println();
    }
    public static void consonant(String str)
    {
        System.out.print("Consonants: ");
        for (int i=0;i<str.length();i++)
        {
            if (Character.isLetter(str.charAt(i)) && "AEIOUaeiou".
            indexOf(str.charAt(i)) == -1)
            {
                System.out.print(str.charAt(i) + " ");
            }
        }
        System.out.println();
    }
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String str = sc.nextLine();
        vowel(str);
        consonant(str);
    }
}
```

4.
```
import java.util.*;
class ReverseString {
    static String rev_string(String str)
    {
```

```
        String rev="";
        for (int i=0;i<str.length();i++)
        {
            rev=str.charAt(i)+rev;
        }
        return rev;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String str = sc.nextLine();
        System.out.println("Reversed String: " + rev_string(str));
    }
}
```

5.
```
import java.util.*;
class PrimeNumbers
 {
    public static void prime(int n)
    {
        System.out.print("Prime numbers up to " + n + ": ");
        for (int i = 2; i <= n; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
        System.out.println();
    }
    public static boolean isPrime(int num)
    {
        if (num < 2) return false;
        for (int i = 2; i <= Math.sqrt(num); i++)
        {
            if (num % i == 0) return false;
```

```
            }
            return true;
        }
        public static void main() {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter a number: ");
            int n = sc.nextInt();
            prime(n);
        }
    }
```

6.
```
    import java.util.*;
    class Rectangle
    {
        static int area(int length, int breadth)
        {
            return length * breadth;
        }
        static int perimeter(int length, int breadth)
        {
            return 2 * (length + breadth);
        }
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            int len,bre;
            System.out.print("Enter length and breadth of the rectangle:
            ");
            len = sc.nextInt();
            bre = sc.nextInt();
            System.out.println("Area: " + area(len, bre));
            System.out.println("Perimeter: " + perimeter(len, bre));
        }
    }
```

7.
```
import java.util.*;
class TimeConverter
{
    public static void time(int seconds)
    {
        int hours = seconds / 3600;
        int minutes = (seconds % 3600) / 60;
        int sec = seconds % 60;
        System.out.println("Time: " + hours + " hours " + minutes +
        " minutes " + sec + " seconds");
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter time in seconds: ");
        int seconds = sc.nextInt();
        time(seconds);
    }
}
```

8.
```
import java.util.*;
class PronicNumber
{
    public static boolean pronic(int num) {
        for (int i = 0; i * (i + 1) <= num; i++)
        {
            if (i * (i + 1) == num)
            return true;
        }
        return false;
    }
    public static void main() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
```

```
        int num = sc.nextInt();
        System.out.println(num + (pronic(num) ? " is a Pronic number"
        : " is not a Pronic number"));
    }
}
```

9.
```
import java.util.*;
class GCDLCM
 {
    public static int GCD(int a, int b) {
        int i,s;
        s=(a<b)?a:b;
        for(i=s;i>=1;i--)
        {
            if(a%i==0 && b%i==0)
                break;
        }
        return i;
    }
    static int LCM(int a, int b) {
        return (a * b) / GCD(a, b);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter two numbers: ");
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println("GCD: " + GCD(a, b));
        System.out.println("LCM: " + LCM(a, b));
    }
}
```

10.
```
import java.util.*;
class TaxCalculator
```

```
{
    public static void cal_tax(double salary)
    {
        double tax = 0;
        if (salary > 1000000)
            tax = 7000 + 0.15 * (salary - 300000);
        else if (salary > 200000)
            tax = 5000 + 0.12 * (salary - 200000);
        else if (salary > 100000)
            tax = 0.05 * (salary - 100000);
        System.out.println("Tax to be paid: Rs. " + tax);
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your salary: ");
        double salary = sc.nextDouble();
        cal_tax(salary);
    }
}
```

11.
```
import java.util.*;
class DiscountCalculator
 {
    public static void discount(double sales)
    {
        double discountRate = 0,discount,discountedPrice;
        if (sales > 10000)
            discountRate = 0.10;
        else if (sales >= 6000)
            discountRate = 0.08;
        else if (sales >= 2000)
            discountRate = 0.04;
```

```
            discount = sales * discountRate;
            discountedPrice = sales - discount;
            System.out.println("Discount: Rs. " + discount);
          System.out.println("Discounted Price: Rs. " + discountedPrice);
        }
        public static void main() {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter sales amount: ");
            double sales = sc.nextDouble();
            discount(sales);
        }
    }
```

12.
```
    import java.util.*;
    class ElectricityBill
    {
        public static void bill(int units)
        {
            double total = 0;
            if (units > 450)
            {
                total += (units - 450) * 15;
                units = 450;
            }
            if (units > 250)
            {
                total += (units - 250) * 10;
                units = 250;
            }
            if (units > 100)
            {
                total += (units - 100) * 8;
                units = 100;
```

```
            }
            total += units * 5;
            System.out.println("Total bill: Rs. " + total);
        }
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter total units consumed: ");
            int units = sc.nextInt();
            bill(units);
        }
    }
```

13.
```
    import java.util.*;
    class AreaCalculator
    {
        public static void area(int length, int breadth)
        {
          System.out.println("Area of Rectangle: " + (length * breadth));
        }
        public static void area(int side)
        {
            System.out.println("Area of Square: " + (side * side));
        }
        public static void area(double radius)
        {
            System.out.println("Area of Circle: " + (Math.PI * radius *
            radius));
        }
        public static void main() {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter length and breadth of rectangle: ");
            int l = sc.nextInt();
```

```
            int b = sc.nextInt();
            area(l, b);
            System.out.print("Enter side of square: ");
            int s = sc.nextInt();
            area(s);
            System.out.print("Enter radius of circle: ");
            double r = sc.nextDouble();
            area(r);
        }
    }
```

14.
```
import java.util.*;
class AdvancedAreaCalculator
 {
    public static double Area(double base, int height)
    {
        return base * height;  // Parallelogram
    }
    public static double Area(double d1, double d2)
    {
        return (d1 * d2) / 2;  // Rhombus
    }
    public static double Area(int a, int b, double h)
    {
        return 0.5 * (a + b) * h;  // Trapezium
    }
    public static void main() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter base and height of parallelogram:
        ");
        double base = sc.nextDouble();
        int height = sc.nextInt();
        System.out.println("Area of Parallelogram: " + Area(base,
        height));
```

```
            System.out.print("Enter diagonals of rhombus: ");
            double d1 = sc.nextDouble();
            double d2 = sc.nextDouble();
            System.out.println("Area of Rhombus: " + Area(d1, d2));
            System.out.print("Enter parallel sides and height of trapezium:
            ");
            int a = sc.nextInt();
            int b = sc.nextInt();
            double h = sc.nextDouble();
            System.out.println("Area of Trapezium: " + Area(a, b, h));
        }
    }
```

15.
```
    import java.util.*;
    class ShapePrinter
     {
        public static void shape(int m, int n)
        {
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    System.out.print("x");
                }
                System.out.println();
            }
        }
        public static void shape(int s, char ch)
        {
            for (int i = 0; i < s; i++)
            {
                for (int j = 0; j < s; j++)
                {
```

```
                System.out.print(ch);
            }
            System.out.println();
        }
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter m and n for rectangle: ");
        int m = sc.nextInt();
        int n = sc.nextInt();
        shape(m, n);
        System.out.print("Enter size and character for square: ");
        int s = sc.nextInt();
        char ch = sc.next().charAt(0);
        shape(s, ch);
    }
}
```

16.
```
import java.util.*;
class WordProcessor {
    static String word(String w) {
        String rw="";
        for(int i=0;i<w.length();i++)
        {
            rw=w.charAt(i)+rw;
        }
        return rw;
    }
    static int word(String w, char ch) {
        int count = 0;
        ch = Character.toLowerCase(ch);
        for (int i=0;i<w.length();i++) {
```

```
            if (w.charAt(i) == ch) count++;
        }
        return count;
    }
    public static void main() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Menu: \n1. Reverse Word\n2. Count
        Occurrences");
        int ch = sc.nextInt();
        switch (ch) {
            case 1:
                System.out.print("Enter word: ");
                String w = sc.next();
                System.out.println("Reversed Word: " + word(w));
                break;
            case 2:
                System.out.print("Enter word: ");
                w = sc.next();
                System.out.print("Enter character to count: ");
                char ch1 = sc.next().charAt(0);
                System.out.println("Occurrences of '" + ch1 + "': "
                + word(w, ch1));
                break;
            default:
                System.out.println("Invalid choice.");
        }
    }
}
```

17.
```
import java.util.*;
class PalindromeChecker {
    static void palindrome(String str) {
        String rw="";
        for(int i=0;i<str.length();i++)
```

```
        {
            rw=str.charAt(i)+rw;
        };
        if (str.equalsIgnoreCase(rw))
            System.out.println(str + " is a Palindromic word.");
        else
            System.out.println(str + " is NOT a Palindromic word.");
    }
    static void palindrome(int num) {
        int temp = num, rev = 0;
        while (temp > 0) {
            rev = rev * 10 + temp % 10;
            temp /= 10;
        }
        if (num == rev)
            System.out.println(num + " is a Palindromic number.");
        else
            System.out.println(num + " is NOT a Palindromic number.");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Menu: \n1. Check Word Palindrome\n2.
        Check Number Palindrome");
        int choice = sc.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter a word: ");
                String word = sc.next();
                palindrome(word);
                break;
            case 2:
                System.out.print("Enter a number: ");
```

```
                int num = sc.nextInt();
                palindrome(num);
                break;
            default:
                System.out.println("Invalid choice.");
        }
        sc.close();
    }
}
```

18.
```
import java.util.*;
public class SeriesCalculator
 {
    public static int series(int n) {
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += Math.pow(i, i);
        }
        return sum;
    }
    public static void series(int x, int n) {
        int sum = 0;
        for (int i = 1; i <= n; i += 2) {
            sum += (x - i);
        }
        System.out.println("Sum of series: " + sum);
    }
    public static void series(double n) {
        System.out.print("Series: ");
        for (int i = 1; i <= n; i++) {
            System.out.print((int)Math.pow(i, 2) + " ");
        }
        System.out.println();
```

```
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Menu:\n1. Sum of Powers Series\n2. Sum
        of Alternating Series\n3. Square Series");
        int choice = sc.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter n: ");
                int n = sc.nextInt();
                System.out.println("Sum: " + series(n));
                break;
            case 2:
                System.out.print("Enter x and n: ");
                int x = sc.nextInt();
                int num = sc.nextInt();
                series(x, num);
                break;
            case 3:
                System.out.print("Enter last term (n): ");
                double last = sc.nextDouble();
                series(last);
                break;
            default:
                System.out.println("Invalid choice.");
        }
    }
}
```

19.
```
import java.util.*;
class OverloadedShow
 {
    public static double show(double n)
```

```
    {
        double sum = 0;
        for (int i = 1; i <= n; i++)
        {
            sum += (double) i / (4 * i);
        }
        return sum;
    }
    public static double show(double a, double n)
    {
        return n / (5 * a);
    }
    public static void main() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Menu: \n1. Show Series 1\n2. Show Series
        2");
        int choice = sc.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter last term (n): ");
                double n = sc.nextDouble();
                System.out.println("Sum: " + show(n));
                break;
            case 2:
                System.out.print("Enter value (a) and last term (n):
                ");
                double a = sc.nextDouble();
                double num = sc.nextDouble();
                System.out.println("Sum: " + show(a, num));
                break;
            default:
                System.out.println("Invalid choice.");
        }
    }
}
```

11. Class as the Basis of all Computation

MIND DRILL

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

B. 1.

```
import java.util.*;
class Employee
 {
    String name;
    double basic, da, hra, pf, np, gp;
    // Method to take input
    void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Name: ");
        name = sc.nextLine();
        System.out.print("Enter Basic Salary: ");
        basic = sc.nextDouble();
    }
    // Method to calculate salary components
    void cal()
    {
        da = 0.25 * basic;  // Dearness Allowance (25% of Basic Pay)
        hra = 0.15 * basic; // House Rent Allowance (15% of Basic
        Pay)
       pf = 0.0833 * basic; // Provident Fund (8.33% of Basic Pay)
        np = basic + da + hra; // Net Pay
        gp = np - pf; // Gross Pay
    }
    // Method to display the calculated salary details
    void display()
    {
        System.out.println("\nEmployee Details:");
        System.out.println("Name: " + name);
```

```
        System.out.println("Basic Salary: " + basic);
        System.out.println("Dearness Allowance (DA): " + da);
        System.out.println("House Rent Allowance (HRA): " + hra);
        System.out.println("Provident Fund (PF): " + pf);
        System.out.println("Net Pay: " + np);
        System.out.println("Gross Pay: " + gp);
    }
    public static void main()
    {
        Employee emp = new Employee();
        emp.input(); // Taking input
        emp.cal();   // Calculating allowances & deductions
        emp.display(); // Displaying the salary details
    }
}
```

2.
```
import java.util.*;
class Tax
 {
    // Data Members
     String name;
     String department;
     double monthlySalary;
     double incomeTax;
    // Method to accept teacher details
    void acceptDetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name: ");
        name = sc.nextLine();
        System.out.print("Enter Department: ");
        department = sc.nextLine();
        System.out.print("Enter Monthly Salary: ");
        monthlySalary = sc.nextDouble();
    }
```

```
    // Method to compute annual income tax
    void computeTax() {
        double annualSalary = monthlySalary * 12;
        if (annualSalary > 200000) {
            incomeTax = 0.12 * (annualSalary - 200000);
        } else {
            incomeTax = 0;
        }
    }
    // Method to display teacher details
    void displayDetails() {
        System.out.println("\nTeacher Details:");
        System.out.println("Name: " + name);
        System.out.println("Department: " + department);
        System.out.println("Monthly Salary: Rs." + monthlySalary);
        System.out.println("Annual Salary: Rs." + (monthlySalary *
        12));
        System.out.println("Income Tax: Rs." + incomeTax);
    }
    public static void main()
    {
        Tax teacher = new Tax();
        teacher.acceptDetails();
        teacher.computeTax();
        teacher.displayDetails();
    }
}
```

3.
```
import java.util.*;
class Bill {
    // Data Members
     String name;
     String cons_no;
     int unit_con;
     double billAmount;
```

```
// Method to accept the data members
void getdata() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter Consumer Name: ");
    name = sc.nextLine();
    System.out.print("Enter Consumer Number: ");
    cons_no = sc.nextLine();
    System.out.print("Enter Units Consumed: ");
    unit_con = sc.nextInt();
}
// Method to compute bill amount
void compute() {
    if (unit_con <= 120) {
        billAmount = unit_con * 1.20;
    } else if (unit_con <= 250) {
        billAmount = (120 * 1.20) + (unit_con - 120) * 2.20;
    } else if (unit_con <= 400) {
        billAmount = (120 * 1.20) + (130 * 2.20) + (unit_con -
        250) * 3.20;
    } else {
        billAmount = (120 * 1.20) + (130 * 2.20) + (150 * 3.20)
        + (unit_con - 400) * 4.00;
    }
}
// Method to display the bill details
void display() {
    System.out.print("\nConsumer Name \nConsumer No \nUnit Consumed
    \nBill Amount\n");
    System.out.print(name+" "+ cons_no+" "+ unit_con+" "+
    billAmount);
}
public static void main() {
    Bill customerBill = new Bill();
    customerBill.getdata();
```

```
            customerBill.compute();
            customerBill.display();
        }
    }
```

4.
```
import java.util.*;
class School_book {
    // Instance variables
    int name;
    String author_name;
    double price_book;
    int number_book;
    // Method to accept data from user
    void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter book name (as an integer): ");
        name = sc.nextInt();
        sc.nextLine(); // Consume newline
        System.out.print("Enter author name: ");
        author_name = sc.nextLine();
        System.out.print("Enter price of the book: ");
        price_book = sc.nextDouble();
        System.out.print("Enter number of books to be bought: ");
        number_book = sc.nextInt();
    }
    // Method to calculate total cost
    void cal_cost() {
        double total_cost = price_book * number_book;
        System.out.println("Total cost: " + total_cost);
    }
    // Method to display details
    void display() {
        System.out.println("Book Name: " + name);
        System.out.println("Author Name: " + author_name);
        System.out.println("Price per Book: " + price_book);
```

```
            System.out.println("Number of Books: " + number_book);
            cal_cost(); // Displaying the total cost as well
        }
        // Main method
        public static void main(String[] args) {
            School_book book = new School_book();
            book.accept();
            book.display();
        }
    }
```

5.
```
    import java.util.*;
    class BookFair
    {
        // Instance variables
        String Bname;
        double price;
        // Method to input data
        void input()
        {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter book name: ");
            Bname = sc.nextLine();
            System.out.print("Enter book price: ");
            price = sc.nextDouble();
        }
        // Method to calculate price after discount
        void calculate()
        {
            double discount = 0;
            if (price <= 1000)
            {
                discount = price * 0.02;
            }
```

```
        else if (price > 1000 && price <= 3000)
        {
            discount = price * 0.10;
        }
        else if (price > 3000)
        {
            discount = price * 0.15;
        }
        price -= discount;
    }
    // Method to display book details
    void display()
    {
        System.out.println("Book Name: " + Bname);
        System.out.println("Price after discount: " + price);
    }
    // Main method
    public static void main()
    {
        BookFair book = new BookFair();
        book.input();
        book.calculate();
        book.display();
    }
}
```

6.
```
import java.util.*;
class RailwayTicket {
    // Instance variables
    String name;
    String coach;
    long mobno;
    int amt;
    int totalamt;
    // Method to accept details
```

```
void accept() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter customer name: ");
    name = sc.nextLine();
    System.out.print("Enter coach type (FirstAC, SecondAC, ThirdAC,
    Sleeper): ");
    coach = sc.nextLine().toLowerCase();
    System.out.print("Enter mobile number: ");
    mobno = sc.nextLong();
    System.out.print("Enter basic ticket amount: ");
    amt = sc.nextInt();
}
// Method to update total amount based on coach type
void update() {
    int extra = 0;
    switch (coach) {
        case "firstac":
            extra = 700;
            break;
        case "secondac":
            extra = 500;
            break;
        case "thirdac":
            extra = 250;
            break;
        case "sleeper":
            extra = 0;
            break;
        default:
            System.out.println("Invalid coach type. Defaulting
            to Sleeper with no extra charge.");
    }
    totalamt = amt + extra;
}
```

```
        // Method to display details
        void display() {
            System.out.println("Customer Name: " + name);
            System.out.println("Coach Type: " + coach);
            System.out.println("Mobile Number: " + mobno);
            System.out.println("Total Amount: " + totalamt);
        }
        // Main method
        public static void main(String[] args) {
            RailwayTicket ticket = new RailwayTicket();
            ticket.accept();
            ticket.update();
            ticket.display();
        }
    }
```

7.
```
    import java.util.*;
    class Discount {
        // Instance variables
        String item;
        double cost;
        double discount;
        double amt;
        // Method to input item name and cost
        void input() {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter item name: ");
            item = sc.nextLine();
            System.out.print("Enter item cost: ");
            cost = sc.nextDouble();
        }
        // Method to calculate discount based on cost
        void cal() {
            if (cost <= 2000) {
                discount = 0;
```

```
        } else if (cost > 2000 && cost <= 5000) {
            discount = cost * 0.10;
        } else if (cost > 5000 && cost <= 7000) {
            discount = cost * 0.15;
        } else {
            discount = cost * 0.20;
        }
        amt = cost - discount;
    }
    // Method to display details
    void display() {
        System.out.println("Item: " + item);
        System.out.println("Price: " + cost);
        System.out.println("Discount: " + discount);
        System.out.println("Final Amount: " + amt);
    }
    // Main method
    public static void main(String[] args) {
        Discount d = new Discount();
        d.input();
        d.cal();
        d.display();
    }
}
```

8.
```
import java.util.*;
class Interest {
    // Data members
    double p; // Principal amount
    double r; // Rate of interest
    int t;    // Time in years
    double si; // Simple Interest
    double amt; // Total amount to be paid
    // Method to input p, r, and t
    void input() {
```

```
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Principal Amount: ");
        p = sc.nextDouble();
        System.out.print("Enter Rate of Interest: ");
        r = sc.nextDouble();
        System.out.print("Enter Time (in years): ");
        t = sc.nextInt();
    }
    // Method to calculate simple interest and final amount
    void calculate() {
        si = (p * r * t) / 100;
        amt = p + si;
    }
    // Method to display Simple Interest and Final Amount
    void show() {
        System.out.println("Simple Interest: " + si);
        System.out.println("Total Amount to be Paid: " + amt);
    }
    // Main method
    public static void main() {
        Interest obj = new Interest();
        obj.input();
        obj.calculate();
        obj.show();
    }
}
```

9.
```
import java.util.Scanner;
class School_Library {
    // Data members
    String name; // Book Name
    int day_late; // Number of days late
    double fine; // Fine amount to be paid
    // Method to accept book name and number of days late
    void accept() {
```

```
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Book Name: ");
        name = sc.nextLine();
        System.out.print("Enter Number of Days Late: ");
        day_late = sc.nextInt();
    }
    // Method to calculate fine
    void calculate() {
        if (day_late <= 5) {
            fine = day_late * 5;
        } else if (day_late <= 10) {
            fine = (5 * 5) + ((day_late - 5) * 7);
        } else if (day_late <= 30) {
            fine = (5 * 5) + (5 * 7) + ((day_late - 10) * 10);
        } else {
            fine = (5 * 5) + (5 * 7) + (20 * 10) + ((day_late - 30)
            * 15);
        }
    }
    // Method to display details
    void display() {
        System.out.println("-----------------------------------------");
        System.out.println("Name of Book | Number of Days Late |
        Total Fine");
        System.out.println("-----------------------------------------");
        System.out.println( name+"  \t\t| "+  day_late +"  \t\t| "
        + fine);
        System.out.println("-----------------------------------------");
    }
    // Main method
    public static void main() {
        School_Library book = new School_Library();
        book.accept();
        book.calculate();
```

```
            book.display();
        }
    }
```

10.
```
import java.util.*;
class Loan
{
    // Data members
    int principal; // Principal amount to be taken
    int time; // Time of repayment
    double rate; // Rate of interest
    // Method to accept principal and time
    void accept()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Principal Amount: ");
        principal = sc.nextInt();
        System.out.print("Enter Time of Repayment (in years): ");
        time = sc.nextInt();
    }
    // Method to calculate simple interest
    double cal_interest()
    {
        if (time == 1)
        {
            rate = 4.5;
        }
        else if (time == 2)
        {
            rate = 5.0;
        }
        else if (time == 3)
        {
            rate = 7.5;
        }
```

```
            else
            {
                rate = 10.0;
            }
            return (principal * rate * time) / 100;
        }
        // Method to display principal and interest calculated
        void display()
        {
            double interest = cal_interest();
            System.out.println("Principal Amount: " + principal);
           System.out.println("Time of Repayment: " + time + " years");
            System.out.println("Rate of Interest: " + rate + "%");
            System.out.println("Simple Interest: Rs. " + interest);
        }
        // Main method
        public static void main()
        {
            Loan loan = new Loan();
            loan.accept();
            loan.display();
        }
    }
```

11.
```
    import java.util.*;
    class Automorphic
     {
        public static void automorphic(int num)
        {
            String numStr,squareStr;
            int square;
            square = num * num;
            numStr = String.valueOf(num);
            squareStr = String.valueOf(square);
            if (squareStr.endsWith(numStr))
```

```
        {
            System.out.println(num + " is an Automorphic number.");
        }
        else
        {
         System.out.println(num + " is not an Automorphic number.");
        }
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int num;
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        automorphic(num);
    }
}
```

12.
```
import java.util.*;
class Palindrome
{
    public static boolean isPalindrome(int num)
    {
        int original = num, reverse = 0;
        while (num > 0)
        {
            reverse = reverse * 10 + num % 10;
            num /= 10;
        }
        return original == reverse;
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int m,n,i;
```

```
            System.out.print("Enter range (m and n): ");
            m = sc.nextInt();
            n = sc.nextInt();
            System.out.println("Palindromic numbers between " + m + "
            and " + n + ":");
            for (i = m; i <= n; i++)
            {
                if (isPalindrome(i))
                {
                    System.out.print(i + " ");
                }
            }
        }
    }
```

13.
```
import java.util.*;
class quadratic
 {
    double a,b,c,r1,r2,d;
    void accept()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the coefficients a, b, and c:");
        a = sc.nextDouble();
        b = sc.nextDouble();
        c = sc.nextDouble();
    }
    void determinant()
    {
        d = b * b - 4 * a * c;
    }
    void root()
    {
        if (d > 0)
        {
```

```
            r1 = (-b + Math.sqrt(d)) / (2 * a);
            r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("The roots are real and distinct:");
            System.out.println("Root 1 = " + r1);
            System.out.println("Root 2 = " + r2);
        }
        else if (d == 0)
        {
            r1 = -b / (2 * a);
            System.out.println("The roots are real and equal:");
            System.out.println("Root = " + r1);
        }
        else
        {
            System.out.println("Roots are imaginary");
        }
    }
    public static void main()
    {
        quadratic ob = new quadratic();
        ob.accept();
        ob.determinant();
        ob.root();
    }
}
```

14.
```
import java.util.*;
class reverse_string
 {
     String str,rstr;
     void accept()
     {
         Scanner sc= new Scanner(System.in);
         System.out.println("Enter string ");
```

```
            str=sc.nextLine();
        }
    void  change()
    {
        rstr="";
        for (int i=0;i<str.length();i++)
        {
            rstr=str.charAt(i)+rstr;
        }
    }
    void show()
    {
        System.out.println(str);
        System.out.println(rstr);
    }
    public static void main() {
        reverse_string ob = new reverse_string();
        ob.accept();
        ob.change();
        ob.show();
    }
}
```

15.
```
import java.util.*;
class word
{
    String w;
    int vc,cc;
    void input()
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter word ");
        w=sc.next();
        vc=0;
        cc=0;
```

```
}
void count_vowels()
{
    int i;
    for(i=0;i<w.length();i++)
    {
        char ch=w.charAt(i);
        if("AEIOUaeiou".indexOf(ch)!=-1)
        {
            vc++;
        }
    }
}
void count_consonants()
{
    int i;
    w=w.toUpperCase();
    for(i=0;i<w.length();i++)
    {
        char ch=w.charAt(i);
        if(ch>='A' && ch<='Z')
        {
            if("AEIOUaeiou".indexOf(ch)==-1)
            {
                cc++;
            }
        }
    }
}
void show()
{
    System.out.println("Vowels "+vc);
    System.out.println("Consonants "+cc);
}
```

```
    public static void main() {
        word ob = new word();
        ob.input();
        ob.count_vowels();
        ob.count_consonants();
        ob.show();
    }
}
```

C. 1. c. A is true, but R is false.

2. a. Both A and R are true, and R is the correct explanation of A.
3. b. Both A and R are true, but R is not the correct explanation of A.
4. a. Both A and R are true, and R is the correct explanation of A.

D. 1. d 2. c 3. d 4. d 5. d (Both b and c)

12. Constructors

MIND DRILL

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

B. 1. A default constructor is a constructor that does not take any parameters. It is used to initialize an object with default values. If a class does not have a constructor, the compiler automatically provides a default constructor.

2. Syntax is :

   ClassName objectName = new ClassName(); // Calls the default constructor

   ClassName objectName = new ClassName(arguments); // Calls a parameterized constructor

3. **Parameterized Constructor:** A constructor that takes parameters to initialize objects with specific values. It is used when different objects need different initial values.

   Non-Parameterized Constructor: A constructor that does not take any parameters and initializes objects with default values. It is used when all objects should have the same default values.

C. 1. c. A is true, but R is false.

2. c. A is true, but R is false.
3. c. A is true, but R is false.

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

Unsolved Programs

1.
```
import java.util.*;
class Employee {
    // Data members
    String ename;
    double basicsal, hra, da, pf, gross, net;
    // Default Constructor
    Employee() {
        ename = "";
        basicsal = hra = da = pf = gross = net = 0.0;
    }
    // Parameterized Constructor
    Employee(String name, double basic) {
        ename = name;
        basicsal = basic;
    }
    // Method to calculate salary components
    void calculate() {
        hra = 0.10 * basicsal;  // 10% of basic salary
        da = 0.55 * basicsal;   // 55% of basic salary
        pf = 0.0833 * basicsal; // 8.33% of basic salary
        gross = basicsal + hra + da;
        net = gross - pf;
    }
    // Method to display all data members
    void display() {
        System.out.println("Employee Name: " + ename);
        System.out.println("Basic Salary: " + basicsal);
        System.out.println("HRA: " + hra);
        System.out.println("DA: " + da);
        System.out.println("PF: " + pf);
        System.out.println("Gross Salary: " + gross);
        System.out.println("Net Salary: " + net);
```

```
    }
    public static void main() {
        Scanner sc = new Scanner(System.in);
        // Taking input from user
        System.out.print("Enter Employee Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Basic Salary: ");
        double salary = sc.nextDouble();
        // Creating Employee Object using Parameterized Constructor
        Employee emp = new Employee(name, salary);
        emp.calculate(); // Calculate salary details
        emp.display();   // Display details
    }
}
```

2.
```
import java.util.*;
class Salesman
{
    // Data Members
    String sname;
    int sales;
    double rate, tot_amt;
    // Non-parameterized constructor
    Salesman()
    {
        sname = "";
        sales = 0;
        rate = 0.0;
        tot_amt = 0.0;
    }
    // Method to take input from user
    void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Salesman Name: ");
```

```
            sname = sc.nextLine();
            System.out.print("Enter Quantity of Sales: ");
            sales = sc.nextInt();
            System.out.print("Enter Rate per Quantity: ");
            rate = sc.nextDouble();
        }
        // Method to calculate total amount
        void compute()
        {
            tot_amt = sales * rate;
        }
        // Method to display details
        void display()
        {
            System.out.println("\nSalesman Details:");
            System.out.println("Name: " + sname);
            System.out.println("Quantity of Sales: " + sales);
            System.out.println("Rate per Quantity: " + rate);
            System.out.println("Total Sales Amount: " + tot_amt);
        }
        public static void main()
        {
            Salesman sm = new Salesman(); // Creating an object of Salesman
            sm.input();    // Taking input
            sm.compute(); // Computing total amount
            sm.display(); // Displaying details
        }
    }
```

3.
```
    import java.util.*;
    class HCF_LCM
    {
        // Data Members
        int a, b;
        int hcf, lcm;
```

```
// Constructor
HCF_LCM()
{
    a = 0;
    b = 0;
    hcf = 0;
    lcm = 0;
}
// Method to take input
void input()
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter two numbers: ");
    a = sc.nextInt();
    b = sc.nextInt();
}
// Method to calculate HCF
void hcf_cal()
{
    int x=Math.max(a,b);
    int i;
    for(i=x;i>=1;i--)
    {
        if(a%i==0 && b%i==0)
        {
            hcf=i;
            break;
        }
    }
    System.out.println("HCF: " + hcf);
}
// Method to calculate LCM
void lcm_cal()
{
```

```
        lcm = (a * b) / hcf; // Using formula LCM * HCF = a * b
        System.out.println("LCM: " + lcm);
    }
    // Main Method
    public static void main()
    {
        HCF_LCM obj = new HCF_LCM(); // Object Creation
        obj.input();  // Taking Input
        obj.hcf_cal(); // Calculating and displaying HCF
        obj.lcm_cal(); // Calculating and displaying LCM
    }
}
```

4.
```
import java.util.*;
class Computer
{
    // Data Members
    String model_name;
    double org_price;
    double inc_amount;
    double total;
    // Non-parameterized constructor
    Computer()
    {
        model_name = "";
        org_price = 0.0;
        inc_amount = 0.0;
        total = 0.0;
    }
    // Parameterized constructor
    Computer(String mn, double op)
    {
        model_name = mn;
        org_price = op;
        calculatePrice();
```

```
}
// Method to calculate the increased price based on given criteria
void calculatePrice()
{
    if (org_price <= 20000)
    {
        inc_amount = org_price * 0.025;
    }
    else if (org_price <= 30000)
    {
        inc_amount = org_price * 0.027;
    }
    else if (org_price <= 40000)
    {
        inc_amount = org_price * 0.04;
    }
    else
    {
        inc_amount = org_price * 0.06;
    }
    total = org_price + inc_amount;
}
// Method to display the data members
void display()
{
    System.out.println("Model Name "+"\t"+" Original Price "+"\t"+
    " Increased Amount " +"\t"+" Total");
    System.out.println(model_name+ " \t "+ org_price +" \t "+
    inc_amount +" \t\t "+ total);
}
public static void main()
{
    // Example usage
    Computer c1 = new Computer("Dell Inspiron", 25000);
```

```
            c1.display();
        }
    }
```

5.
```
import java.util.*;
class Palindrome1
{
    // Data Members
    String word;
    String revword;
    // Constructor
    Palindrome1()
    {
        word = "";
        revword = "";
    }
    // Method to take input
    void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a word: ");
        word = sc.nextLine();
    }
    // Method to check if the word is a palindrome
    boolean check()
    {
        int i;
        for(i=0;i<word.length();i++)
        {
            revword = word.charAt(i) + revword;
        }
        return word.equalsIgnoreCase(revword);
    }
    // Method to display the result
    void display()
```

```
    {
        if (check())
        {
            System.out.println(word + " is a Palindrome.");
        }
        else
        {
            System.out.println(word + " is NOT a Palindrome.");
        }
    }
    // Main method for execution
    public static void main()
    {
        Palindrome1 p = new Palindrome1();
        p.input();
        p.display();
    }
}
```

6.
```
import java.util.*;
class Encode
{
    // Data Members
    String word;
    String encodeword;
    // Non-parameterized Constructor
    Encode()
    {
        word = "";
        encodeword = "";
    }
    // Parameterized Constructor
    Encode(String w)
    {
```

```
        word = w;
        encodeword = "";
    }
    // Method to convert and encode the word
    void convert()
    {
        encodeword = ""; // Reset encoded word
        for (int i = 0; i < word.length(); i++)
        {
            char ch = word.charAt(i);
            if (ch == 'Z')
            {
                encodeword += 'A'; // If 'Z', replace with 'A'
            }
            else if (ch == 'z')
            {
                encodeword += 'a'; // If 'z', replace with 'a'
            }
            else
            {
                encodeword += (char) (ch + 1); // Replace with next
                ASCII character
            }
        }
        System.out.println("Encoded Word: " + encodeword);
    }
    // Main method to test the class
    public static void main()
    {
        String s;
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter word ");
        s=sc.next();
        Encode e1 = new Encode(s); // Example input
```

```
            e1.convert(); // Encoding the word
        }
    }
```

7.
```
import java.util.*;
class Pronic
 {
    // Data Member
    int n;
    // Constructor to initialize n to 0
    Pronic()
    {
        n = 0;
    }
    // Method to input n from the user
    void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        n = sc.nextInt();
    }
    // Method to check if the number is a Pronic number
    void check()
    {
        boolean isPronic = false;
        for (int i = 1; i<n; i++)
        {
            if (i * (i + 1) == n)
            {
                isPronic = true;
                break;
            }
        }
        // Display the result
        if (isPronic)
```

```
            {
                System.out.println(n + " is a Pronic Number.");
            }
            else
            {
                System.out.println(n + " is NOT a Pronic Number.");
            }
        }
        // Main method to execute the program
        public static void main()
        {
            Pronic p = new Pronic(); // Creating object
            p.input(); // Taking input
            p.check(); // Checking and displaying result
        }
    }
```

8.
```
    class Series
    {
        // Data Members
        double sum;
        double n;
        double x;
        // Constructor
        Series() {
            sum = 0.0;
            n = 0.0;
            x = 0.0;
        }
        // Parameterized Constructor
        Series(double x1, double x2)
        {
            x = x1;
            n = x2;
            sum = 0.0;
```

```
    }
    // Method to calculate the sum
    void calculate()
    {
        for (int i = 1; i <= (int)n; i++)
        {
            sum += x / Math.pow(i, 2);
        }
    }
    // Method to display the result
    void display()
    {
        System.out.println("The sum of the series is: "+ sum);
    }
    // Main method for testing
    public static void main()
    {
        Series s = new Series(2, 4);  // Example: x = 2, n = 4
        s.calculate();
        s.display();
    }
}
```

9.
```
class pattern
 {
    int n;
    // Constructor to initialize n
    pattern(int n1)
    {
        n = n1;
    }
    void pattern1()
    {
        for (int i = 1; i <= n; i++)
        {
```

```
            for (int j = 1; j <= i; j++)
            {
                System.out.print(j);
            }
            System.out.println();
        }
    }
    void pattern2()
    {
        int num = 1;
        for (int i = 1; i <= n; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                System.out.print(num + " ");
                num += 2;
            }
            System.out.println();
        }
    }
    // Example usage
    public static void main()
    {
        pattern obj = new pattern(5);
        System.out.println("Pattern 1:");
        obj.pattern1();
        System.out.println("Pattern 2:");
        obj.pattern2();
    }
}
```

10.
```
import java.util.*;
class fare
{
    int dis;          // dis traveled
```

```
double bill;      // Bill amount
// Default constructor
fare()
{
    dis = 0;
    bill = 0.0;
}
// Parameterized constructor
fare(int d)
{
    dis = d;
    bill = 0.0;
}
// Method to calculate fare
void calculate()
{
    if (dis <= 10)
    {
        bill = dis * 5;
    }
    else if (dis <= 30)
    {
        bill = 10 * 5 + (dis - 10) * 7.5;
    }
    else if (dis <= 50)
    {
        bill = 10 * 5 + 20 * 7.5 + (dis - 30) * 10;
    }
    else
    {
        bill = 10 * 5 + 20 * 7.5 + 20 * 10 + (dis - 50) * 15;
    }
}
// Method to display data members
```

```
    void display()
    {
        System.out.println("distance Travelled: " + dis + " km");
        System.out.println("Total Bill: ₹" + bill);
    }
    // Example usage
    public static void main()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter displance travelled : ");
        int d=sc.nextInt();
        fare f = new fare(d); // example dis
        f.calculate();
        f.display();
    }
}
```

13. Library Classes

MIND DRILL

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

B. 1. true 2. lowercase 3. converting String to double
4. Integer.parseInt() method. 5. char data type.

C. 1. E:true
2. 39.6139.6
3. parseInt() converts a string to an integer.
   toString() converts a value (like an int) to its string representation.
4. a. Float.parseFloat() – Converts a string into a float.
   b. Character.isDigit() – Checks if the character is a digit.
5. A wrapper class is an object representation of a primitive type.
   Example: Integer is the wrapper class for int.

6. Autoboxing: Automatically converting a primitive type to its wrapper class.

Unboxing: Automatically converting a wrapper class to its primitive type.

Non-Parameterized Constructor: A constructor that does not take any parameters and initializes objects with default values. It is used when all objects should have the same default values.

D. 1. c 2. a 3. d

E. 1. i. b ii. b iii. b

2. i. b ii. a iii. c

14. Encapsulation And Inheritance

MIND DRILL

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

6. a 7. c 8. c 9. a 10. b

B. 1. subclass

2. Modular programming

3. hybrid inheritance

4. subclass

C. 1. c 2. a 3. a

D. 1. i. b ii. b iii. b

2. i. b ii. c iii. c

15. Arrays

MIND DRILL

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

6. d 7. b 8. b 9. c 10. b

B. 1.
```
import java.util.*;
class prog
{
public static void main()
```

```
{
int arr[]  = new int[10];
Scanner sc = new Scanner(System.in);
int sum = 0, product = 1;
for(int i = 0; i < 10; i++) {
    arr[i] = sc.nextInt();
    if(arr[i] % 6 == 0) sum += arr[i];
    if(arr[i] < 0) product *= arr[i];
}
System.out.println("Sum of multiples of 6: " + sum);
System.out.println("Product of negative numbers: " + product);
}
}
```

2.
```
import java.util.*;
class palin_array
{
    boolean isPalindrome(int num)
    {
        int original = num, reverse = 0;
        while(num != 0)
        {
            reverse = reverse * 10 + num % 10;
            num /= 10;
        }
        return original == reverse;
    }
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int arr[]=new int[25];
        int i;
        boolean b;
        for(i=0;i<25;i++)
        {
```

```
            System.out.println("Enter a number : ");
            arr[i]=sc.nextInt();
        }
        palin_array ob= new palin_array();
        System.out.println(" The palindromes are :");
        for(i=0;i<25;i++)
        {
            b=ob.isPalindrome(arr[i]);
            if(b)
                System.out.print(arr[i] + " ");
        }
    }
}
```

3.
```
import java.util.*;
class array3
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int arr[]=new int[10];
        int i,sum=0;
        for(i=0;i<10;i++)
        {
            System.out.println("Enter a number : ");
            arr[i]=sc.nextInt();
            if(i % 2 == 1 && arr[i] % 2 == 0)
                sum += arr[i];
        }
        System.out.println("Sum = " + sum);
    }
}
```

4.
```
import java.util.*;
class array3
{
```

```
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int[] A = new int[5];
        int[] B = new int[5];
        int[] C = new int[5];
        System.out.println("Enter elements in A array ");
        for(int i = 0; i < 5; i++)
        {
            A[i] = sc.nextInt();
        }
        System.out.println("Enter elements in B array ");
        for(int i = 0; i < 5; i++)
        {
            B[i] = sc.nextInt();
            C[i] = A[i] + B[i];
        }
        System.out.print("C array : ");
        for(int i = 0; i < 5; i++)
        {
            System.out.print(C[i]+" ");
        }
    }
}
```

5.
```
import java.util.*;
class array4
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int[] arr = new int[10];
        int i,pos;
        for(i = 0; i < 10; i++)
        {
```

```
            arr[i] = sc.nextInt();
        }
        System.out.println("Enter position to delete:");
        pos = sc.nextInt();
        for(i = pos; i < 9; i++)
        {
            arr[i] = arr[i + 1];
        }
        System.out.println("-------------------------");
        for(i = 0; i < 10; i++)
        {
            System.out.print(arr[i]+" ");
        }
    }
}
```

6.
```
import java.util.*;
class array4
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        String[] items = new String[5];
        int[] prices = new int[5];
        int total = 0,i;
        System.out.println("Enter item names and prices :");
        for(i = 0; i < 5; i++)
        {
            items[i] = sc.next();
            prices[i] = sc.nextInt();
            total += prices[i];
        }
        System.out.println("Item Names "+ "\t"+ "prices ");
        for(i = 0; i < 5; i++)
        {
```

```
            System.out.println(items[i]+" \t\t "+ prices[i] );
        }
        System.out.println("Total Amount = " + total);
    }
}
```

7.
```
import java.util.*;
class array4
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int[] arr = new int[10];
        int i,value,pos;
        System.out.println("Enter the elements in the array ");
        for(i = 0; i < 9; i++)
            arr[i] = sc.nextInt();
        System.out.println("Enter the value to be entered ");
        value = sc.nextInt();
        System.out.println("Enter the position to be entered ");
        pos = sc.nextInt();
        for(i = 8; i >= pos; i--)
            arr[i + 1] = arr[i];
        arr[pos] = value;
        System.out.println("The elements in the array ");
        for(i = 0; i < 10; i++)
            System.out.print(arr[i]+" ");
    }
}
```

8.
```
import java.util.*;
class array4
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
```

```
        String names[] = new String[10];
        int i,j,min;
        String temp;
        System.out.println("Enter the names ");
        for(i = 0; i < 10; i++)
            names[i] = sc.next();
        for(i = 0; i < names.length - 1; i++)
        {
            min = i;
            for(j = i + 1; j < names.length; j++)
            {
                if(names[j].compareTo(names[min]) < 0) min = j;
            }
            temp = names[i];
            names[i] = names[min];
            names[min] = temp;
        }
        System.out.println("Enter the names ");
        for(i = 0; i < 10; i++)
            System.out.print(names[i]+" ");
    }
}
```

9.
```
import java.util.*;
class array4
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        int n,j=0,k=0,i;
        System.out.println("Enter the size of the array ");
        n=sc.nextInt();
        int arr[] = new int[n];
        int even[] = new int[n];
        int odd[] = new int[n];
```

```
            for(i = 0; i < n; i++)
            {
                arr[i] = sc.nextInt();
                if(arr[i] % 2 == 0)
                    even[j++]=arr[i];
                else
                    odd[k++]=arr[i];
            }
            System.out.print("\nEven Numbers are : ");
            for(i = 0; i < j; i++)
            {
                System.out.print(even[i]+" ");
            }
            System.out.print("\nOdd Numbers are : ");
            for(i = 0; i < k; i++)
            {
                System.out.print(odd[i]+" ");
            }
        }
    }
```

10.
```
    import java.util.*;
    class array4
    {
        public static void main()
        {
            Scanner sc= new Scanner(System.in);
            int sum = 0,temp,digitSum,i;
            int num[]=new int[20];
            for(i = 0; i < 20; i++)
            {
                num[i] = sc.nextInt();
            }
            for(i = 0; i < 20; i++)
            {
```

```
                temp = num[i];
                digitSum = 0;
                while(temp != 0)
                {
                    digitSum += temp % 10;
                    temp /= 10;
                }
                if(num[i] % digitSum == 0)
                    sum += num[i];
            }
            System.out.println("Sum of Niven numbers: " + sum);
        }
    }
```

11.
```
import java.util.*;
class Student
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String names[]  = new String[40];
        int marks[] = new int[40];
        String tempn;
        int i,j,tempm;
        // Accept student data
        for (i = 0; i < 40; i++)
        {
            System.out.print("Enter name of student " + (i + 1) + ":
            ");
            names[i] = sc.nextLine();
            System.out.print("Enter marks in Computer Science: ");
            marks[i] = sc.nextInt();
        }
        // Sort using bubble sort (descending order of marks)
        for (i = 0; i < 40 - 1; i++)
```

```
            {
                for (j = 0; j < 40 - i - 1; j++)
                {
                    if (marks[j] < marks[j + 1])
                    {
                        // Swap
                        tempm = marks[j];
                        marks[j] = marks[j + 1];
                        marks[j + 1] = tempm;
                        tempn = names[j];
                        names[j] = names[j + 1];
                        names[j + 1] = tempn;
                    }
                }
            }
            // Display sorted list
            System.out.println("\n--- Students Sorted by Merit ---");
            for (i = 0; i < 40; i++)
            {
                System.out.println(names[i] + " - " + marks[i]);
            }
        }
    }
```

12.
```
    import java.util.*;
    class MaxMin
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            int numbers[] = new int[25];
            int i,max,min;
            // Input 25 elements
            System.out.println("Enter 25 numbers:");
            for (i = 0; i < 25; i++)
```

```
        {
            System.out.print("Number " + (i + 1) + ": ");
            numbers[i] = sc.nextInt();
        }
        // Initialize max and min with the first element
        max = numbers[0];
        min = numbers[0];
        // Traverse the array to find max and min
        for (i = 1; i < 25; i++)
        {
            if (numbers[i] > max)
            {
                max = numbers[i];
            }
            if (numbers[i] < min)
            {
                min = numbers[i];
            }
        }
        // Display results
        System.out.println("\nMaximum value: " + max);
        System.out.println("Minimum value: " + min);
    }
}
```

13.
```
import java.util.*;
class ReverseNumbers
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int numbers[] = new int[15];
        int i,num,rev,digit;
        // Input 15 numbers
        System.out.println("Enter 15 numbers:");
```

```
        for (i = 0; i < 15; i++)
        {
            System.out.print("Number " + (i + 1) + ": ");
            numbers[i] = sc.nextInt();
        }
        // Reverse each number and replace in the array
        for (i = 0; i < 15; i++)
        {
            num=numbers[i] ;
            rev = 0;
            while (num != 0)
            {
                digit = num % 10;
                rev = rev * 10 + digit;
                num = num / 10;
            }
            numbers[i]=rev;
        }
        // Display the reversed numbers
        System.out.println("\nArray after reversing each number:");
        for (i = 0; i < 15; i++)
        {
         System.out.println("Element " + (i + 1) + ": " + numbers[i]);
        }
    }
}
```

14.
```
import java.util.*;
class CommonElements
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int[] A = new int[5];
        int[] B = new int[5];
```

```
        int i,j;
        // Input for array A
        System.out.println("Enter 5 elements for array A:");
        for (i = 0; i < 5; i++)
        {
            A[i] = sc.nextInt();
        }
        // Input for array B
        System.out.println("Enter 5 elements for array B:");
        for (i = 0; i < 5; i++)
        {
            B[i] = sc.nextInt();
        }
        // Find and print common elements
        System.out.println("\nCommon elements:");
        for (i = 0; i < 5; i++)
        {
            for (j = 0; j < 5; j++)
            {
                if (A[i] == B[j] && A[i]!=-999)
                {
                    System.out.println(A[i]);
                    A[i] = -999; // Prevents printing duplicates if
                    B has repeated values
                }
            }
        }
    }
}
```

15.
```
import java.util.*;
class StudentAverage
{
    public static void main()
    {
```

```
Scanner sc = new Scanner(System.in);
int[] roll = new int[50];
int[] subA = new int[50];
int[] subB = new int[50];
int[] subC = new int[50];
double[] avg = new double[50];
// Input marks
for (int i = 0; i < 50; i++)
{
    roll[i] = i + 1; // Roll numbers start from 1
  System.out.println("Enter marks for Roll No. " + roll[i]);
    System.out.print("Subject A: ");
    subA[i] = sc.nextInt();
    System.out.print("Subject B: ");
    subB[i] = sc.nextInt();
    System.out.print("Subject C: ");
    subC[i] = sc.nextInt();
    avg[i] = (subA[i] + subB[i] + subC[i]) / 3.0;
}
System.out.println("\nAverage marks of each student:");
for (int i = 0; i < 50; i++) {
    System.out.print("Roll No. "+ roll[i] +" Average : "+
    avg[i]);
}
System.out.println("\nStudents with average > 80:");
for (int i = 0; i < 50; i++) {
    if (avg[i] > 80) {
        System.out.print("Roll No. "+ roll[i] +" Average :
        "+ avg[i]);
    }
}
System.out.println("\nStudents with average < 40:");
for (int i = 0; i < 50; i++) {
    if (avg[i] < 40) {
```

```
                System.out.print("Roll No. "+ roll[i] +" Average :
                "+ avg[i]);
            }
        }
    }
}
```

16.
```
import java.util.Scanner;
public class MatrixDiagonals {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[][] matrix = new int[3][3];
        int mainDiagonalSum = 0;
        int secondaryDiagonalSum = 0;
        // Input values into 3x3 matrix
        System.out.println("Enter 9 numbers for a 3x3 matrix:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print("Element [" + i + "][" + j + "]:
                ");
                matrix[i][j] = sc.nextInt();
            }
        }
        // Display matrix
        System.out.println("\nMatrix:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(matrix[i][j] + "\t");
            }
            System.out.println();
        }
        // Display main diagonal and calculate sum
        System.out.print("\nMain Diagonal: ");
        for (int i = 0; i < 3; i++) {
            System.out.print(matrix[i][i] + " ");
```

```
                mainDiagonalSum += matrix[i][i];
            }
          System.out.println("\nSum of Main Diagonal: " + mainDiagonalSum);
            // Display secondary diagonal and calculate sum
            System.out.print("\nSecondary Diagonal: ");
            for (int i = 0; i < 3; i++) {
                System.out.print(matrix[i][2 - i] + " ");
                secondaryDiagonalSum += matrix[i][2 - i];
            }
            System.out.println("\nSum of Secondary Diagonal: " +
            secondaryDiagonalSum);
        }
    }
```

17.
```
import java.util.*;
  class SumProduct1
   {
      public static void main()
      {
          Scanner sc = new Scanner(System.in);
          int arr[][] = new int[4][3];  // 4 rows, 3 columns
          int i,j,rowSum,colPro;
          // Accepting array elements
          System.out.println("Enter 12 numbers for a 4x3 array:");
          for ( i = 0; i < 4; i++) {
              System.out.println("Row " + (i + 1) + ":");
              for (j = 0; j < 3; j++) {
                   System.out.print("Element [" + i + "][" + j + "]:
");
                  arr[i][j] = sc.nextInt();
              }
          }
          // Sum of each row
          System.out.println("\nSum of each row:");
          for ( i = 0; i < 4; i++) {
```

```
                rowSum = 0;
                for ( j = 0; j < 3; j++) {
                    rowSum += arr[i][j];
                }
               System.out.println("Row " + (i + 1) + " Sum: " + rowSum);
            }
            // Product of each column
            System.out.println("\nProduct of each column:");
            for ( j = 0; j < 3; j++)
            {
                colPro = 1;
                for ( i = 0; i < 4; i++)
                {
                    colPro *= arr[i][j];
                }
                System.out.println("Column " + (j + 1) + " Product: " +
                colPro);
            }
        }
    }
```

18.
```
import java.util.*;
  class ColSort
   {
      public static void main()
      {
          Scanner sc = new Scanner(System.in);
          int matrix[][] = new int[2][4];
          int i,j,temp;
          // Input matrix elements
          System.out.println("Enter 8 numbers for a 2x4 matrix:");
          for (i = 0; i < 2; i++)
          {
              for (j = 0; j < 4; j++)
              {
```

```
                System.out.print("Element [" + i + "][" + j + "]:
                ");
                matrix[i][j] = sc.nextInt();
            }
        }
        // Sort each column (only 2 rows, so just compare and swap
        if needed)
        for (j = 0; j < 4; j++)
        {
            if (matrix[0][j] > matrix[1][j])
            {
                // Swap values
                temp = matrix[0][j];
                matrix[0][j] = matrix[1][j];
                matrix[1][j] = temp;
            }
        }
        // Display sorted matrix
        System.out.println("\nMatrix after sorting each column:");
        for (i = 0; i < 2; i++)
        {
            for (j = 0; j < 4; j++)
            {
                System.out.print(matrix[i][j] + "\t");
            }
            System.out.println();
        }
    }
}
```

19.
```
import java.util.*;
class Symmetric
 {
    public static void main()
    {
```

```
Scanner sc = new Scanner(System.in);
int matrix[][] = new int[3][3];
int i,j;
// Input matrix elements
System.out.println("Enter 9 elements for a 3x3 matrix:");
for (i = 0; i < 3; i++)
{
    for (j = 0; j < 3; j++)
    {
        System.out.print("Enter value : ");
        matrix[i][j] = sc.nextInt();
    }
}
// Display the matrix
System.out.println("\nThe matrix is:");
for ( i = 0; i < 3; i++) {
    for (j = 0; j < 3; j++) {
        System.out.print(matrix[i][j] + "\t");
    }
    System.out.println();
}
// Check for symmetry
boolean isSymmetric = true;
for ( i = 0; i < 3; i++)
{
    for ( j = 0; j < 3; j++)
    {
        if (matrix[i][j] != matrix[j][i])
        {
            isSymmetric = false;
            break;
        }
    }
    if (!isSymmetric)
```

```
                break;
            }
            // Output result
            if (isSymmetric)
            {
                System.out.println("\nThe matrix is symmetric.");
            } else
            {
                System.out.println("\nThe matrix is NOT symmetric.");
            }
            sc.close();
        }
    }
```

20.
```
class Diagonal
  {
      public static void main()
      {
          // Example matrices
          int[][] A = {
              {1, 2, 3},
              {4, 5, 6},
              {7, 8, 9}
          };
          int[][] B = {
              {9, 8, 7},
              {6, 5, 4},
              {3, 2, 1}
          };
          // Get the sum of left diagonal of matrix A
          int sumLeft = 0,n = A.length,i,sumRight=0;  // Assuming square
          matrix
          for (i = 0; i < n; i++)
          {
```

```
            sumLeft += A[i][i];
        }
        // Get the sum of right diagonal of matrix B
        for ( i = 0; i < n; i++) {
            sumRight += B[i][n - 1 - i];
        }
        // Print the result
        System.out.println("Sum of left diagonal of matrix A: " +
        sumLeft);
        System.out.println("Sum of right diagonal of matrix B: " +
        sumRight);
        System.out.println("Result (Sum of left diagonal of A / Sum
        of right diagonal of B): "
                            + (double) sumLeft/ sumRight);
    }
}
```

21.
```
import java.util.*;
class EvenNum
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int matrix[][] = new int[2][3];
        int i,j;
        // Input elements
        System.out.println("Enter 6 numbers for a 2x3 matrix:");
        for ( i = 0; i < 2; i++)
        {
            System.out.println("Row " + (i + 1) + ":");
            for ( j = 0; j < 3; j++)
            {
                System.out.print("Enter number in array : ");
                matrix[i][j] = sc.nextInt();
            }
```

```
            }
            // Display even numbers row-wise
            System.out.println("\nEven numbers row-wise:");
            for (i = 0; i < 2; i++)
            {
                System.out.print("Row " + (i + 1) + ": ");
                boolean hasEven = false;
                for (j = 0; j < 3; j++)
                {
                    if (matrix[i][j] % 2 == 0)
                    {
                        System.out.print(matrix[i][j] + " ");
                        hasEven = true;
                    }
                }
                if (!hasEven) {
                    System.out.print("No even numbers");
                }
                System.out.println();
            }
        }
    }
```

22.
```
    import java.util.*;
    class Pascals
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            int n,i,j;
            // Ask the user for the number of rows in Pascal's Triangle
            System.out.print("Enter the number of rows for Pascal's
            Triangle: ");
            n = sc.nextInt();
           // Declare a 2D array to hold the values of Pascal's Triangle
```

```
        int triangle[][] = new int[n][n];
        // Generate Pascal's Triangle
        for (i = 0; i < n; i++)
        {
            // First and last elements in each row are always 1
            triangle[i][0] = 1;
            triangle[i][i] = 1;
            // Calculate the values in between using the formula:
            // triangle[i][j] = triangle[i-1][j-1] + triangle[i-1]
            [j]
            for (j = 1; j < i; j++)
            {
                triangle[i][j] = triangle[i-1][j-1] + triangle[i-1]
                [j];
            }
        }
        // Print Pascal's Triangle
        System.out.println("Pascal's Triangle:");
        for (i = 0; i < n; i++)
        {
            // Print spaces for formatting the triangle
            for (j = 0; j < n - i - 1; j++)
            {
                System.out.print(" ");
            }
            // Print the elements of the row
            for (j = 0; j <= i; j++) {
                System.out.print(triangle[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

23.
```
import java.util.*;
class Largest
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Declare a 4x4 matrix
        int arr[][] = new int[4][4];
        int i,j,max;
        // Accept numbers in the array
        System.out.println("Enter the elements of a 4x4 matrix:"5);
        for (i = 0; i < 4; i++)
        {
            for (j = 0; j < 4; j++)
            {
                arr[i][j] = sc.nextInt();
            }
        }
        System.out.println("Matrix Format ");
        for (i = 0; i < 4; i++)
        {
            for (j = 0; j < 4; j++)
            {
                System.out.print(arr[i][j] +"\t");
            }
            System.out.println();
        }
        // Find the largest number column-wise
        for (j = 0; j < 4; j++)
        {
             max = arr[0][j];  // Assume first element in the column
             is the largest
            for ( i = 1; i < 4; i++)
            {
```

```
                    if (arr[i][j] > max)
                    {
                        max = arr[i][j];  // Update max if a larger number
                        is found
                    }
                }
                System.out.println("Largest number in column " + (j + 1)
                + ": " + max);
            }
        }
    }
```

16. String Handling

MIND DRILL

A.

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

B.

1. String
2. concatenation
3. String
4. leading and trailing whitespaces
5. valueOf()

C. 1.

```
import java.util.*;
class Q1
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a sentence:");
        String str = sc.nextLine();
        int countA = 0, vowels = 0, consonants = 0,len;
        len = str.length();
        for (int i = 0; i <len; i++)
```

```
        {
                char ch = str.charAt(i);
                if(ch=='A')
                    countA++;
                if (Character.isLetter(ch))
                {
                    if ("AEIOUaeiou".indexOf(ch) != -1)
                        vowels++;
                    else
                        consonants++;
                }
            }
            System.out.println("Words starting with A: " + countA);
            System.out.println("Vowels: " + vowels);
            System.out.println("Consonants: " + consonants);
        }
    }
```

2.
```
    import java.util.*;
    class Q2
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter str:");
            String str = sc.nextLine();
            System.out.println("Enter character:");
            char ch = sc.next().charAt(0);
            str = str.toUpperCase();
            char target = Character.toUpperCase(ch);
            int count = 0;
            String newstr = "";
            for (int i = 0; i < str.length(); i++)
            {
                char current = str.charAt(i);
```

```
                if (current == target)
                {
                    count++;
                    newstr += "*";
                }
                else
                {
                    newstr += current;
                }
            }
            System.out.println("Number of times '" + ch + "' present: "
+ count);
            System.out.println("New str: " + newstr);
        }
    }
```

3.
```
import java.util.*;
class Q3
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine()+" ",longest = "",word="";
        int i,len;
        char ch;
        len=str.length();
        for (i=0;i<len;i++)
        {
            ch=str.charAt(i);
            if(ch!=' ')
                word+=ch;
            else
            {
                if (word.length() > longest.length())
                {
```

```
                    longest = word;
                }
                word="";
            }
        }
        System.out.println("The longest word: " + longest);
    }
}
```

4.
```
import java.util.*;
class Q4
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine()+" ",smallest = str ,word="";
        int i,len;
        char ch;
        len=str.length();
        for (i=0;i<len;i++)
        {
            ch=str.charAt(i);
            if(ch!=' ')
                word+=ch;
            else
            {
                if (word.length() < smallest.length())
                {
                    smallest = word;
                }
                word="";
            }
        }
        System.out.println("The samllest word: " + smallest);
```

```
    }
}
```

5.
```
import java.util.*;
class Q5
{
    public static void main()
{
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a non-palindromic word:");
        String word = sc.nextLine(),reversed = "";
        int i;
        for (i = word.length() - 1; i >= 0; i--)
     {
            reversed += word.charAt(i);
        }
        word +=reversed;
        System.out.println("The palindromic word is: " + word);
    }
}
```

6.
```
import java.util.*;
class Q6
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        for (int i = 0; i < str.length(); i++)
        {
            char ch = str.charAt(i);
            System.out.println(ch + "=" + (int) ch);
        }
    }
}
```

7.
```
import java.util.*;
class Q7
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String fName,fIn,mIn,lN;
        int sp1,sp2;
        System.out.println("Enter full name:");
        fName = sc.nextLine().trim();
        sp1 = fName.indexOf(' ');
        sp2 = fName.lastIndexOf(' ');
        fIn = fName.substring(0, sp1).trim().toUpperCase().charAt(0)
        + ".";
        mIn = fName.substring(sp1 + 1, sp2).trim().toUpperCase().
        charAt(0) + ".";
        lN = fName.substring(sp2 + 1);
        System.out.println("Output: " + lN + ", " + fIn + mIn);
    }
}
```

8.
```
import java.util.*;
class Q8
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String str;
        int i,count;
        char ch;
        System.out.println("Enter a sentence :");
        str = sc.nextLine().toUpperCase();
        str = str.replace(" ", "");
        while (str.length() > 0)
        {
```

```
                ch = str.charAt(0);
                count = 0;
                for(i = 0; i < str.length(); i++) {
                    if (str.charAt(i) == ch)
                        count++;
                }
                System.out.println(ch + " = " + count);
                str = str.replace((ch+""), "");
            }
        }
    }
```

9.
```
import java.util.*;
class q9
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a sentence ");
        String str = sc.nextLine()+" ",word="",lgword="";
        int i,len,c1=0,c2=0;
        char ch;
        len=str.length();
        for(i=0;i<len;i++)
        {
            ch=str.charAt(i);
            if(ch!=' ')
            {
                word+=ch;
                if("AEIOUaeiou".indexOf(ch)!=-1)
                {
                    c1++;
                }
            }
            else
```

```
                {
                    if (c1>c2)
                    {
                        lgword = word;
                        c2=c1;
                    }
                    word="";
                    c1=0;
                }
            }
            System.out.println("The word with largest number of vowels
: " + lgword);
            System.out.println("Number of vowels present : " + c2);
        }
    }
```

10.
```
import java.util.*;
class reversewordinstring
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a sentence ");
        String str = sc.nextLine()+" ",word="",nstr="";
        int i,len;
        char ch;
        len=str.length();
        for(i=0;i<len;i++)
        {
            ch=str.charAt(i);
            if(ch!=' ')
            {
                word=ch+word;
            }
            else
```

```
            {
                nstr=nstr+" "+word;
                word="";
            }
        }
        System.out.println("Original Sentence : " + str);
        System.out.println("Reversed sentence : " + nstr);
    }
}
```

11.
```
import java.util.*;
class UniqueWord
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        String word ;
        boolean unique = true;
        int i,len;
        char ch;
        System.out.println("Enter a word ");
        word=sc.next().toUpperCase();
        len=word.length();
        for (i = 0; i < len; i++)
        {
            ch = word.charAt(i);
            if (word.indexOf(ch) != word.lastIndexOf(ch)) {
                unique = false;
                break;
            }
        }
        if (unique)
            System.out.println(word + " is a unique word.");
        else
            System.out.println(word + " is not a unique word.");
```

```
        }
    }
```

12.
```
import java.util.*;
class replacevowels
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String word,nword="";
        System.out.println("Enter a word ");
        word = sc.next();
        int i,len;
        char ch;
        len=word.length();
        for (i = 0; i < len; i++)
        {
            ch = word.charAt(i);
            if ("aeiouAEIOU".indexOf(ch) != -1)
            {
                nword += (char)(ch + 1);
            }
            else
            {
                nword += ch;
            }
        }
        System.out.println(nword);
    }
}
```

13.
```
class RemoveVowelsDigits
{
    public static void main()
    {
        String input = "Pincode of Sodepur is Kolkata - 700115";
```

```
        String vowels = "aeiouAEIOU";
        String result = "";
        for (int i = 0; i < input.length(); i++)
        {
            char ch = input.charAt(i);
            if (vowels.indexOf(ch) == -1 && !Character.isDigit(ch))
            {
                result += ch;
            }
        }
        System.out.println("Output: " + result);
    }
}
```

14.
```
class replace
{
    public static void main()
    {
        String sentence = "India is my country";
        sentence = sentence.replace("my", "our");
        System.out.println(sentence);
    }
}
```

16.
```
import java.util.*;
class repeated_word
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String sentence = sc.nextLine().toUpperCase(); // Convert to
        uppercase for uniform comparison
      String[] words = sentence.split("\\s+"); // Split by whitespace
        boolean[] visited = new boolean[words.length];
        System.out.print("The repeated words are: ");
```

```
        boolean found = false;
        for (int i = 0; i < words.length; i++)
        {
            if (!visited[i])
            {
                int count = 1;
                for (int j = i + 1; j < words.length; j++)
                {
                    if (words[i].equals(words[j]))
                    {
                        count++;
                        visited[j] = true; // Mark repeated word
                    }
                }
                if (count > 1) {
                    System.out.print(words[i] + " ");
                    found = true;
                }
            }
        }
        if (!found) {
            System.out.print("None");
        }
    }
}
```

17.
```
import java.util.*;
class palin_finder
{
    // Check if a word is a palindrome
    public static boolean isPalindrome(String word)
    {
        int left = 0, right = word.length() - 1;
        while (left < right)
        {
```

```
            if (word.charAt(left) != word.charAt(right))
                return false;
            left++;
            right--;
        }
        return true;
    }
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String sentence = sc.nextLine().toUpperCase() + " "; // Add
        space to process the last word
        String word = "";
        String smallest = "";
        String largest = "";
        for (int i = 0; i < sentence.length(); i++)
        {
            char ch = sentence.charAt(i);
            if (ch != ' ')
            {
                word += ch; // Build current word
            }
            else
            {
                if (word.length() > 0 && isPalindrome(word))
                {
                        if (smallest.equals("") || word.length() <
smallest.length())
                    {
                        smallest = word;
                    }
                 if (largest.equals("") || word.length() > largest.
length())
```

```
                        {
                            largest = word;
                        }
                    }
                    word = ""; // Reset word after space
                }
            }
            if (!smallest.equals("") && !largest.equals(""))
            {
                System.out.println("Smallest palindromic word: " +
smallest);
             System.out.println("Largest palindromic word: " + largest);
            } else {
                System.out.println("No palindromic words found.");
            }
        }
    }
```

18.
```
import java.util.*;
  class Words_Same_Start_End
  {
      public static void main()
      {
          Scanner sc = new Scanner(System.in);
          String word = "",str;
          int len,i;
          char ch;
          // Input a str
          System.out.println("Enter a str:");
          str = sc.nextLine().trim() + " "; // Extra space to handle
last word
          str= str.toUpperCase();
          len = str.length();
          System.out.println("Words that start and end with the same
          letter:");
```

```
            for (i = 0; i < len; i++)
            {
                    ch = str.charAt(i);
                if (ch != ' ')
                {
                    word += ch; // Build the current word
                }
                else
                {
                    // Only process non-empty words
                    if (word.length() > 0)
                    {
                        char first = word.charAt(0);
                        char last = word.charAt(word.length() - 1);
                        if (first == last)
                        {
                            System.out.println(word);
                        }
                    }
                    word = ""; // Reset for next word
                }
            }
        }
    }
```

19.
```
class City
  {
      // Check if character is a vowel (A, E, I, O, U)
      public static boolean isVowel(char ch) {
          ch = Character.toUpperCase(ch);
          return (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' ||
          ch == 'U');
      }
      // Check if character is a consonant
      public static boolean isConsonant(char ch) {
```

```
            ch = Character.toUpperCase(ch);
            return (ch >= 'A' && ch <= 'Z') && !isVowel(ch);
        }
        public static void main()
        {
            // Declare and initialize array with city names
            String[] cities = {"DELHI", "KOLKATA", "JAMSHEDPUR", "LUCKNOW",
            "PATNA"};
            int i,len;
            // Print matching cities
            System.out.println("Cities that begin with a consonant and
            end with a vowel:");
            for (i = 0; i < cities.length; i++)
            {
                len = 0;
                // Get first and last character manually
                char first = cities[i].charAt(0);
                char last = cities[i].charAt(cities[i].length() - 1);
                // Check if first is consonant and last is vowel
                if (isConsonant(first) && isVowel(last)) {
                    System.out.println(cities[i]);
                }
            }
        }
    }
```

20.
```
import java.util.*;
class Names
{
    public static void main()
    {
        Scanner sc= new Scanner(System.in);
        // Initialize array with 10 names
        String[] names = new String[10];
        String temp;
```

```
        int i,j;
        for(i=0;i<10;i++)
        {
            System.out.print("Enter name : ");
            names[i]=sc.next();
        }
        // Sort using simple bubble sort logic
        for (i = 0; i < names.length - 1; i++)
        {
            for (j = 0; j < names.length - 1 - i; j++)
            {
                if (names[j].compareTo(names[j + 1]) > 0)
                {
                    // Swap
                    temp = names[j];
                    names[j] = names[j + 1];
                    names[j + 1] = temp;
                }
            }
        }
        // Print sorted names
        System.out.println("Names in alphabetical order:");
        for (i = 0; i < names.length; i++)
        {
            System.out.print(names[i]+ " ");
        }
    }
}
```

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

```
        // Input a word
        System.out.print("Enter a word: ");
        String word = sc.nextLine();
        int i,j;
        int len = word.length();
        // Print pattern
        for (i = 0; i < len; i++)
        {
            for (j = 0; j < len-i; j++)
            {
                System.out.print(word.charAt(j));
            }
            System.out.println();
        }
    }
}
```

ii.
```
import java.util.*;
class DecreasingPattern
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Input a word
        System.out.print("Enter a word: ");
        String word = sc.nextLine();
        int i,j;
        int len = word.length();
        // Print pattern
        for (i = 0; i < len; i++)
        {
            for (j = 0; j <= i; j++)
            {
                System.out.print(word.charAt(j));
            }
```

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

iii.
```
import java.util.*;
class DecreasingPattern
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Input a word
        System.out.print("Enter a word: ");
        String word = sc.nextLine();
        int i,j;
        int len = word.length();
        // Print pattern
        for (i = 0; i < len; i++)
        {
            for (j = i; j < len ; j++)
            {
                System.out.print(word.charAt(j));
            }
            System.out.println();
        }
    }
}
```

iv.
```
import java.util.*;
class DecreasingPattern
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Input a word
        System.out.print("Enter a word: ");
```

```
            String word = sc.nextLine();
            int i,j;
            int len = word.length();
            // Print pattern
            for (i = 0; i < len; i++)
            {
                for (j = len-1; j >=i ; j--)
                {
                    System.out.print(word.charAt(j));
                }
                System.out.println();
            }
        }
    }
```

22. a.
```
    import java.util.*;
    class DecreasingPattern
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            // Input a word
            System.out.print("Enter a word: ");
            String word = sc.nextLine();
            int i,j;
            int len = word.length();
            // Print pattern
            for (i = len-1; i >= 0; i--) {
                for (j = 0; j <= i; j++) {
                    System.out.print(word.charAt(j));
                }
                System.out.println();
            }
        }
    }
```

b.
```
import java.util.*;
class DecreasingPattern
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Input a word
        System.out.print("Enter a word: ");
        String word = sc.nextLine();
        int i,j;
        int len = word.length();
        // Print pattern
        for (i = 0; i < len; i++) {
            for (j = i; j < len; j++) {
                System.out.print(word.charAt(j));
            }
            System.out.println();
        }
    }
}
```
c.
```
import java.util.*;
class DecreasingPattern
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Input a word
        System.out.print("Enter a word: ");
        String word = sc.nextLine();
        int i,j;
        int len = word.length();
        // Print pattern
        for (i = 0; i < len; i++) {
            for (j = 0; j <= i; j++) {
```

```
                System.out.print(word.charAt(j));
            }
            System.out.println();
        }
    }
}
```

d.
```
import java.util.*;
class DecreasingPattern
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        int i,j,k=0;
        // Print pattern
        for (i = 0; i < 4; i++)
        {
            for (j = 0; j <= i; j++)
            {
                System.out.print((char)(65+k));
                k++;
            }
            System.out.println();
        }
    }
}
```

23.
```
import java.util.*;
class Student_1
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        // Arrays for names and marks
        String[] names = new String[5];
        int[] marks = new int[5];
```

```
int i,j,minIndex,tempM;
String tempN;
// Input 5 student names and marks
for (i = 0; i < 5; i++)
{
    System.out.print("Enter name of student " + (i + 1) + ":
    ");
    names[i] = sc.nextLine();
    System.out.print("Enter marks of " + names[i] + ": ");
    marks[i] = sc.nextInt();
    sc.nextLine(); // Consume newline
}
// Selection Sort based on names (alphabetical)
for (i = 0; i < 4; i++)
{
    minIndex = i;
    for (j = i + 1; j < 5; j++)
    {
        if (names[j].compareTo(names[minIndex]) < 0)
        {
            minIndex = j;
        }
    }
    // Swap names
    tempN = names[i];
    names[i] = names[minIndex];
    names[minIndex] = tempN;
    // Swap corresponding marks
    tempM = marks[i];
    marks[i] = marks[minIndex];
    marks[minIndex] = tempM;
}
// Output sorted list
System.out.println("\nStudents in alphabetical order:");
```

```
            System.out.println("Name"+"\t\t"+"Marks");
            for (i = 0; i < 5; i++)
            {
                System.out.println(names[i] + " \t\t " + marks[i]);
            }
        }
    }
```

24.
```
    import java.util.*;
    class Pattern
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            // Input word
            System.out.print("Enter a word: ");
            String word = sc.nextLine();
            // Calculate length manually
            int len = 0,ch,i,j;// Menu
            System.out.println("Enter choice:");
          System.out.println("1 - Print from full word to last letter");
          System.out.println("2 - Print from last letter to full word");
            ch = sc.nextInt();
            len=word.length();
            if (ch == 1)
            {
                // From full word to last letter
                for (i = 0; i < len; i++)
                {
                    for (j = i; j < len; j++)
                    {
                        System.out.print(word.charAt(j));
                    }
                    System.out.println();
                }
```

```
        }
        else if (ch == 2)
        {
            // From last letter to full word
            for (i = len - 1; i >= 0; i--)
            {
                for (j = i; j < len; j++)
                {
                    System.out.print(word.charAt(j));
                }
                System.out.println();
            }
        }
        else
        {
            System.out.println("Invalid choice.");
        }
    }
}
```

25.
```
import java.util.*;
class shift
 {
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String str;
        int i;
        char ch;
        // Input str
        System.out.print("Enter a str: ");
        str= sc.nextLine();
        System.out.println("Converted str:");
        // Process each character
        for (i = 0;i<str.length();i++)
```

```
        {
                ch = str.charAt(i);
                // If uppercase letter
                if (ch >= 'A' && ch <= 'Z')
                {
                    ch = (char) ('A' + (ch - 'A' + 2) % 26);
                }
                // If lowercase letter
                else if (ch >= 'a' && ch <= 'z')
                {
                    ch = (char) ('a' + (ch - 'a' + 2) % 26);
                }
                // Else keep character as is (space, digit, punctuation,
                etc.)
                System.out.print(ch);
        }
    }
}
```

26.
```
import java.util.*;
class Abbrebiation
{
    public static void main(String str)
    {
        Scanner sc = new Scanner(System.in);
        str=" "+str;
        String abb = "";
        // Flag to detect start of a word
        boolean isWordStart = true;
        int i;
        char ch,ch1;
        // Manually build acronym using first letters
        for(i = 0; i<str.length(); i++)
        {
            ch = str.charAt(i);
```

```
            if (ch == ' ')
            {
                ch1=str.charAt(i+1);
                // Convert to uppercase if needed
                if (ch >= 'a' && ch <= 'z')
                {
                    ch1 = (char) (ch1 - 32); // to uppercase
                }
                abb += ch1;
                isWordStart = false;
            }
        }
        System.out.println("Abbrebiation: " + abb);
    }
}
```

27.
```
import java.util.*;
class frequency
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        String str;
        int i,j,len,charc=0,maxfre=0;
        char ch,maxch='a';
        // Input str
        System.out.print("Enter a str: ");
        str = sc.nextLine().toUpperCase();
        len=str.length();
        // Count character frequencies
        for (i = 65; i <= 90; i++)
        {
            for(j=0;j<len;j++)
            {
                ch = str.charAt(j);
```

```
                    if(ch==(char)i)
                    {
                        charc++;
                    }
                }
                if(maxfre<charc)
                {
                    maxch=(char)i;
                    maxfre=charc;
                }
                charc=0;
                 }
            // Output result
            System.out.println(maxfre + " Character that occurs maximum
            number of times: " + maxch);
        }
    }
```

28.
```
import java.util.*;
  class characterdelete
   {
      public static void main(String str)
      {
        String result = "";  // To store final string with no duplicates
          int i,j;
          char ch;
          str=str.toUpperCase();
          // Loop through each character
          for ( i = 0; i < str.length(); i++)
          {
              ch = str.charAt(i);
              // Check if this character appeared before in 'result'
              boolean repeated = false;
              for (j = 0; j < result.length(); j++)
              {
```

```
                if (result.charAt(j) == ch)
                {
                    repeated = true;
                    break;
                }
            }
            // If not repeated, add to result
            if (!repeated)
            {
                result += ch;
            }
        }
        // Output result
        System.out.println("After removing repeated characters: " +
        result);
    }
}
```

29.
```
import java.util.*;
class charactercommoninwords
 {
    public static void main(String wd1,String wd2)
    {
      String result = "";  // To store final string with no duplicates
        int i,j;
        char ch;
        wd1=wd1.toUpperCase();
        wd2=wd2.toUpperCase();
        // Loop through each character
        for ( i = 0; i < wd1.length(); i++)
        {
            ch = wd1.charAt(i);
            // Check if this character appeared before in 'result'
            boolean repeated = false;
            for (j = 0; j < wd2.length(); j++)
```

```
                {
                    if (wd2.charAt(j) == ch)
                    {
                        repeated = true;
                       wd2 = wd2.substring(0,j)+"*"+wd2.substring(j+1);
                        break;
                    }
                }
                // If not repeated, add to result
                if (repeated)
                {
                    result += ch+",";
                }
            }
            // Output result
            System.out.println("Repeated characters: " + result.
            substring(0,result.length()-1));
        }
    }
```

30.
```
    import java.util.*;
    class WordFre
    {
        public static void main()
        {
            Scanner sc = new Scanner(System.in);
            // Input str
            System.out.print("Enter a str: ");
            String str = sc.nextLine() + " "; // Add space to process
            last word
            String[] words = new String[50]; // max 50 words
            int wordCount = 0;
            String word = "";
            // Extract words manually
            for (int i = 0; i < str.length(); i++) {
```

```
            char ch = str.charAt(i);
            if (ch != ' ')
            {
                word += ch;
            }
            else
            {
                if (word.length() > 0)
                {
                    words[wordCount] = word;
                    wordCount++;
                    word = "";
                }
            }
        }
        // Count frequency of each word
        System.out.println("Word Frequencies:");
        for (int i = 0; i < wordCount; i++)
        {
            if (words[i] == null)
                continue;
            int count = 1;
            for (int j = i + 1; j < wordCount; j++)
            {
                if (words[i].equals(words[j]))
                {
                    count++;
                    words[j] = null; // Mark as counted
                }
            }
            System.out.println(words[i] + " : " + count);
        }
    }
}
```

D. 1. c 2. c 3. a

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