PROGRAMMING IN JAVA

BCA-DS-402

Manav Rachna International Institute of Research and Studies

School of Computer Applications

Department of Computer Applications

Submitted By	
Student Name	**Hardik Linzara**
Roll No	**23/SCA/BCA(DS&BDA)/013**
Programme	**Bachelor of Computer Applications**
Semester	**4th Semester**
Section/Group	**4 - E**
Department	**School of Computer Applications**
Batch	**2023-26**
Submitted To	
Faculty Name	**Dr. Priyanka Sharma**

MANAV RACHNA
vidyanatariksha
Manav Rachna International
Institute of Research and Studies
Deemed to be University under section
3 of the UGC Act, 1956

SCHOOL OF COMPUTER APPLICATIONS

INDEX

22	Generate Prime Numbers in a Range	
23	Simple Class with Methods	
24	Class with Parameterized Constructor	
25	Find Area of Rectangle	
26	Bank Account with Deposit & Withdraw Methods	
27	Method Overloading	
28	Static Methods Demonstration	
29	Method Overriding	
30	Getters and Setters	
31	Class with Multiple Methods	
32	Object Passing in Methods	
33	Area & Perimeter using Super & This	
34	Counting Objects using Static Members	
35	Abstract Classes & Methods	
36	Multilevel Inheritance	
37	Multiple Inheritance using Interfaces	

1. Write a java program to print hello world.

Ans.)
```
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Output
```
Hello World!

=== Code Execution Successful ===
```

2. Java Program to take input from the user and print the sum of two numbers.

Ans.)
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter two numbers");
        int a=scanner.nextInt();
        int b=scanner.nextInt();
        int sum=a+b;
          System.out.println("Sum is: " + sum);
    }
}
```

Output
```
Enter two numbers
5
9
Sum is: 14

=== Code Execution Successful ===
```

3. Create a java program to check whether a number entered by the user is even or odd.

Ans.)
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter a number");
        int a=scanner.nextInt();
        if(a%2==0)
          System.out.println(a+" is even");
          else
          System.out.println(a+" is odd");
    }
}
```

Output

```
Enter a number
8
8 is even

=== Code Execution Successful ===
```

4. Create a java program to print the average and sum of 5 numbers entered by the user.

Ans.)
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter 5 numbers");
        int a=scanner.nextInt();
        int b=scanner.nextInt();
        int c=scanner.nextInt();
        int d=scanner.nextInt();
        int e=scanner.nextInt();
        int sum=a+b+c+d+e;
        int average=sum/5;
        System.out.println("Sum is: "+sum);
         System.out.println("Average is: "+average);
    }
}
```

Output

```
Enter 5 numbers
7
6
```

5. Program to calculate the factorial of a number.

Ans.)
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        int factorial=1;
        System.out.println("Enter a number");
        int a=scanner.nextInt();
        for(int i=a;i>=1;i--)
    {
        factorial=factorial*i;
    }
         System.out.println("Factorial is: "+factorial);
    }
}
```

Output

```
Enter a number
5
Factorial is: 120

=== Code Execution Successful ===
```

6. Program to print Fibonacci series up to n terms.

Ans.)
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        int a=0,b=1,c;
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter a number");
        int n=scanner.nextInt();
          System.out.print(a);
              System.out.print(b);
        for(int i=3;i<=n;i++)
    {
         c=a+b;
```

```
        System.out.print(c);
        a=b;
        b=c;
    }
    }
}
```

Output

```
Enter a number
8
011235813
=== Code Execution Successful ===
```

7. Program to reverse a number.

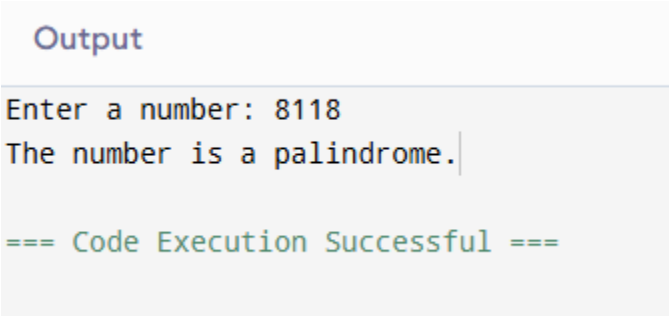
Ans.)
```
import java.util.Scanner;
public class ReverseNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int reversed = 0;
        while (number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number /= 10;
     }
        System.out.println("Reversed Number: " + reversed);
    }
}
```

Output

```
Enter a number: 78925
Reversed Number: 52987

=== Code Execution Successful ===
```

8.) Program to check if a number is palindrome.

Ans.)
```
import java.util.Scanner;
public class PalindromeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int original = number;
        int reversed = 0;
        while (number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number /= 10;
        }
        if (original == reversed) {
            System.out.println("The number is a palindrome.");
        } else {
            System.out.println("The number is not a palindrome.");
        }
    }
}
```



9.) Program for a simple calculator.

Ans.)
```
import java.util.Scanner;
public class SimpleCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter first number: ");
        double num1 = scanner.nextDouble();
        System.out.println("Enter an operator (+, -, *, /): ");
        char operator = scanner.next().charAt(0);
        System.out.println("Enter second number: ");
        double num2 = scanner.nextDouble();
        double result;
        switch (operator) {
            case '+':
                result = num1 + num2;
```

```
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
        case '/':
            if (num2 != 0) {
                result = num1 / num2;
            } else {
                System.out.println("Division by zero is not allowed.");
                scanner.close();
                return;
            }
            break;
        default:
            System.out.println("Invalid operator.");
            scanner.close();
            return;
    }
    System.out.println("The result is: " + result);
  }
}
```

Output

```
Enter first number:
8
Enter an operator (+, -, *, /):
/
Enter second number:
15
The result is: 0.5333333333333333

=== Code Execution Successful ===
```

10.) Program to check if a number is prime.

Ans.)
```
import java.util.Scanner;
public class PrimeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        boolean isPrime = true;
```

```
        if (number <= 1) {
            isPrime = false;
        } else {
            for (int i = 2; i <= Math.sqrt(number); i++) {
                if (number % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }
        if (isPrime) {
            System.out.println("The number is prime.");
        } else {
            System.out.println("The number is not prime.");
        }
    }
}
```

Output

```
Enter a number: 5
The number is prime.

=== Code Execution Successful ===
```

11.) Program to check if a number is an armstrong number.

Ans.)
```
import java.util.Scanner;
public class ArmstrongNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int original = number;
        int sum = 0;
        while (number != 0) {
            int digit = number % 10;
            sum += Math.pow(digit, 3);
            number /= 10;
        }
        if (sum == original) {
            System.out.println("The number is an Armstrong number.");
        } else {
            System.out.println("The number is not an Armstrong number.");
        }
```

```
    }
}
```

Output

```
Enter a number: 153
The number is an Armstrong number.

=== Code Execution Successful ===
```

12.) Program to find the largest of two numbers using ternary operators.

Ans.)
```
import java.util.Scanner;
public class LargestOfTwo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();

        int largest = (num1 > num2) ? num1 : num2;
        System.out.println("The largest number is: " + largest);
    }
}
```

Output

```
Enter the first number: 5
Enter the second number: 7
The largest number is: 7

=== Code Execution Successful ===
```

13.) Program to print multiplication table.

Ans.)
```
import java.util.Scanner;
public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        for (int i = 1; i <= 10; i++) {
            System.out.println(number + " x " + i + " = " + (number * i));
        }
    }
}
```

Output

```
Enter a number: 12
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
12 x 4 = 48
12 x 5 = 60
12 x 6 = 72
12 x 7 = 84
12 x 8 = 96
12 x 9 = 108
12 x 10 = 120

=== Code Execution Successful ===
```

14.) Program to calculate sum and average of array elements.

Ans.)
```
import java.util.Scanner;
public class SumAndAverage {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] array = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            array[i] = scanner.nextInt();
        }
        int sum = 0;
        for (int i = 0; i < n; i++) {
            sum += array[i];
        }
        double average = (double) sum / n;
```

```
        System.out.println("Sum of array elements: " + sum);
        System.out.println("Average of array elements: " + average);
    }
}
```

Output

```
Enter the number of elements in the array: 5
Enter the elements of the array:
4
6
7
8
9
Sum of array elements: 34
Average of array elements: 6.8

=== Code Execution Successful ===
```

15.) Program to reverse a string.

Ans.)
```
import java.util.Scanner;
public class ReverseString {
  public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        String reversed = "";
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed += input.charAt(i);
        }
        System.out.println("Reversed String: " + reversed);
    }
}
```

Output

```
Enter a string: manavrachna
Reversed String: anhcarvanam

=== Code Execution Successful ===
```

16.) Program to find factorial of a number using recursion.

Ans.)
```
import java.util.Scanner;
public class FactorialRecursion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        long factorial = findFactorial(number);
        System.out.println("Factorial of " + number + " is: " + factorial);
    }
    public static long findFactorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        return n * findFactorial(n - 1);
    }
}
```

Output

```
Enter a number: 8
Factorial of 8 is: 40320

=== Code Execution Successful ===
```

17.) Program to sort an array in ascending order.

Ans.)
```
import java.util.Scanner;
import java.util.Arrays;
public class SortArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] array = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            array[i] = scanner.nextInt();
        }
        Arrays.sort(array);
        System.out.println("Array in ascending order:");
        for (int num : array) {
            System.out.print(num + " ");
        }
    }
}
```

Output

```
Enter the number of elements in the array: 5
Enter the elements of the array:
2
8
9
7
4
Array in ascending order:
2 4 7 8 9
=== Code Execution Successful ===
```

18.) Program to check palindrome for a string.

Ans.)
```
import java.util.Scanner;
public class PalindromeString {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String reversed = new StringBuilder(input).reverse().toString();

        if (input.equalsIgnoreCase(reversed)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}
```

Output

```
Enter a string: mvmvmvm
The string is a palindrome.

=== Code Execution Successful ===
```

19.) Program to count vowels and consonants in a string.

Ans.)
```
import java.util.Scanner;
public class VowelConsonantCount {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine().toLowerCase();
        int vowelCount = 0;
        int consonantCount = 0;
        for (char c : input.toCharArray()) {
            if (c >= 'a' && c <= 'z') {
                if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
                    vowelCount++;
                } else {
                    consonantCount++;
                }
            }
        }
        System.out.println("Number of vowels: " + vowelCount);
        System.out.println("Number of consonants: " + consonantCount);
    }
}
```

Output

```
Enter a string: manavrachna
Number of vowels: 4
Number of consonants: 7

=== Code Execution Successful ===
```

20.) Program to implement a simple banking system.

Ans.)
```
import java.util.Scanner;
public class SimpleBankingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double balance = 0.0;
        while (true) {
            System.out.println("\n--- Banking System ---");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Check Balance");
            System.out.println("4. Exit");
```

```
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter the amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    if (depositAmount > 0) {
                        balance += depositAmount;
                        System.out.println("Amount deposited successfully. Current balance: " + balance);
                    } else {
                        System.out.println("Invalid deposit amount.");
                    }
                    break;

                case 2:
                    System.out.print("Enter the amount to withdraw: ");
                    double withdrawalAmount = scanner.nextDouble();
                    if (withdrawalAmount > 0 && withdrawalAmount <= balance) {
                        balance -= withdrawalAmount;
                        System.out.println("Amount withdrawn successfully. Current balance: " + balance);
                    } else if (withdrawalAmount > balance) {
                        System.out.println("Insufficient balance.");
                    } else {
                        System.out.println("Invalid withdrawal amount.");
                    }
                    break;

                case 3:
                    System.out.println("Your current balance is: " + balance);
                    break;

                case 4:
                    System.out.println("Thank you!");
                    scanner.close();
                    return;

                default:
                    System.out.println("Invalid option. Please choose again.");
            }
        }
    }
}
```

Output

```
--- Banking System ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 1
Enter the amount to deposit: 400
Amount deposited successfully. Current balance: 400.0

--- Banking System ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 3
Your current balance is: 400.0

--- Banking System ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 2
Enter the amount to withdraw: 200
Amount withdrawn successfully. Current balance: 200.0

--- Banking System ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 3
Your current balance is: 200.0
```

21.) Program to demonstrate type casting.

Ans.)
```
public class TypeCastingDemo {
    public static void main(String[] args) {
        int intValue = 42;
        double doubleValue = intValue;
        System.out.println("Implicit Type Casting (int to double): " + doubleValue);
        double originalDouble = 42.99;
        int narrowedInt = (int) originalDouble;
        System.out.println("Explicit Type Casting (double to int): " + narrowedInt);
        int num1 = 10;
        int num2 = 3;
        double result = (double) num1 / num2;
        System.out.println("Type Casting in Expressions (int to double): " + result);
    }
}
```

Output

```
Implicit Type Casting (int to double): 42.0
Explicit Type Casting (double to int): 42
Type Casting in Expressions (int to double): 3.3333333333333335

=== Code Execution Successful ===
```

22.) Program to generate prime numbers between 1 and a given number.

Ans.)
```
import java.util.Scanner;
public class PrimeNumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int limit = scanner.nextInt();

        System.out.println("Prime numbers between 1 and " + limit + ":");
        for (int i = 2; i <= limit; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
    }
    private static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
```

```
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

Output

```
Enter a number: 30
Prime numbers between 1 and 30:
2 3 5 7 11 13 17 19 23 29
=== Code Execution Successful ===
```

23.) Program to demonstrate a simple class with methods.

Ans.)
```
public class newclass {
    public int add(int num1, int num2) {
        return num1 + num2;
    }

    public void exit() {
        System.out.println("Exit");
    }

    public static void main(String[] args) {
        newclass newclass = new newclass();
        int sum = newclass.add(10, 20);
        System.out.println("The sum of 10 and 20 is: " + sum);
        newclass.exit();
    }
}
```

Output

```
The sum of 10 and 20 is: 30
Exit

=== Code Execution Successful ===
```

24.) Program for a class with parameterized constructor.

Ans.)
```
public class ParameterizedConstructor {
    String name;
    int age;
    public ParameterizedConstructor(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
    public static void main(String[] args) {
        ParameterizedConstructor person = new ParameterizedConstructor("Chandler", 25);
        person.displayInfo();
    }
}
```

Output

```
Name: Chandler
Age: 25

=== Code Execution Successful ===
```

25.) Program to find area of rectangle using methods.

Ans.)
```
import java.util.Scanner;
public class RectangleArea {
    public double calculateArea(double length, double width) {
        return length * width;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the length of the rectangle: ");
        double length = scanner.nextDouble();
        System.out.print("Enter the width of the rectangle: ");
        double width = scanner.nextDouble();
        RectangleArea rectangle = new RectangleArea();
        double area = rectangle.calculateArea(length, width);
        System.out.println("Area of the rectangle: " + area);
    }
}
```

Output

```
Enter the length of the rectangle: 6
Enter the width of the rectangle: 8
Area of the rectangle: 48.0

    Code Execution Successful
```

26.) Program for Bank account class with deposit and withdraw methods.

Ans.)
```
public class BankAccount {
    private double balance;
    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Successfully deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Successfully withdrawn: " + amount);
        } else if (amount > balance) {
            System.out.println("Insufficient balance.");
        } else {
            System.out.println("Invalid withdrawal amount.");
        }
    }
    public double getBalance() {
        return balance;
    }
    public static void main(String[] args) {
        BankAccount account = new BankAccount(1000.0);
        account.deposit(500.0);
        System.out.println("Current balance: " + account.getBalance());
        account.withdraw(300.0);
        System.out.println("Current balance: " + account.getBalance());
        account.withdraw(1500.0);
        System.out.println("Current balance: " + account.getBalance());
    }
}
```

Output

```
Successfully deposited: 500.0
Current balance: 1500.0
Successfully withdrawn: 300.0
Current balance: 1200.0
Insufficient balance.
Current balance: 1200.0

=== Code Execution Successful ===
```

27.) Program to demonstrate method overloading.

Ans.)
```
public class MethodOverloading {
    public void display(int number) {
        System.out.println("Integer: " + number);
    }
    public void display(String text) {
        System.out.println("String: " + text);
    }
    public void display(double value) {
        System.out.println("Double: " + value);
    }
    public static void main(String[] args) {
        MethodOverloading example = new MethodOverloading();
        example.display(25);
        example.display("Hello, World!");
        example.display(12.34);
    }
}
```

Output

```
Integer: 25
String: Hello, World!
Double: 12.34

=== Code Execution Successful ===
```

28.) Program to demonstrate static methods.

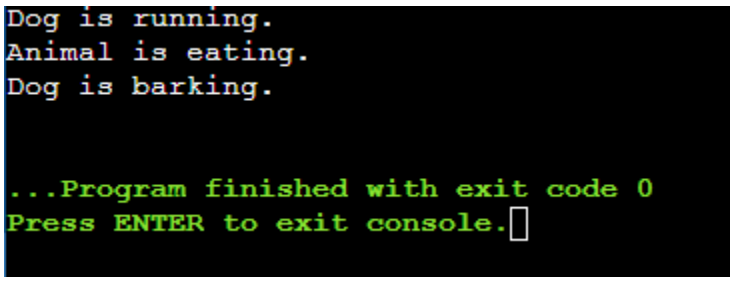
Ans.)
```
public class StaticMethod{
    public static int addNumbers(int a, int b) {
        return a + b;
    }
    public static void displayMessage() {
        System.out.println("Static method called");
    }
    public static void main(String[] args) {
        int sum = StaticMethod.addNumbers(10, 20);
        System.out.println("Sum of 10 and 20 is: " + sum);

        StaticMethod.displayMessage();
    }
}
```

Output

```
Sum of 10 and 20 is: 30
Static method called

=== Code Execution Successful ===
```

29.) Program to demonstrate method overriding.

Ans.)
```
class Animal {
    public void move() {
        System.out.println("Animal is moving.");
    }
    public void eat() {
        System.out.println("Animal is eating.");
    }
}
class Dog extends Animal {
    @Override
    public void move() {
        System.out.println("Dog is running.");
    }
    public void bark() {
        System.out.println("Dog is barking.");
    }
}
public class Main {
    public static void main(String[] args) {
        Dog d = new Dog();
        d.move();
        d.eat();
        d.bark();
    }
}
```



30.) Program to demonstrate getters and setters.

Ans.)
```
public class Person {
    private String name;
    private int age;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
```

```
    }
    public void setAge(int age) {
        if (age > 0) {
            this.age = age;
        } else {
            System.out.println("Age must be a positive number.");
        }
    }
    public static void main(String[] args) {
        Person person = new Person();
        person.setName("Chandler");
        person.setAge(25);
        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());
    }
}
```

Output

```
Name: Chandler
Age: 25

=== Code Execution Successful ===
```

31.) Program to demonstrate class with multiple methods.

Ans.)
```
public class Person {
    private String name;
    private int age;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        if (age > 0) {
            this.age = age;
        } else {
            System.out.println("Age must be a positive number.");
        }
    }
    public static void main(String[] args) {
        Person person = new Person();
```

```
        person.setName("Chandler");
        person.setAge(25);
        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());
    }
}
```

```
Output

Name: Chandler
Age: 25

=== Code Execution Successful ===
```

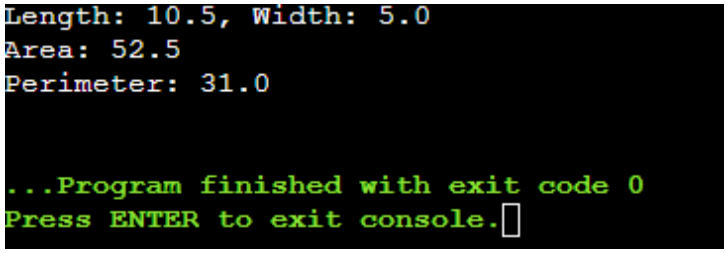
32.) Program to demonstrate object passing in methods.

Ans.)
```
class Rectangle {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double calculateArea() {
        return length * width;
    }
    public void displayDimensions() {
        System.out.println("Length: " + length + ", Width: " + width);
    }
}
public class ObjectPassing {
    public void printRectangleDetails(Rectangle rectangle) {
        rectangle.displayDimensions();
        System.out.println("Area: " + rectangle.calculateArea());
    }
    public static void main(String[] args) {
        Rectangle rect = new Rectangle(10.5, 5.0);
        ObjectPassing objPass = new ObjectPassing();

        objPass.printRectangleDetails(rect);
    }
}
```

```
Length: 10.5, Width: 5.0
Area: 52.5


...Program finished with exit code 0
Press ENTER to exit console.
```

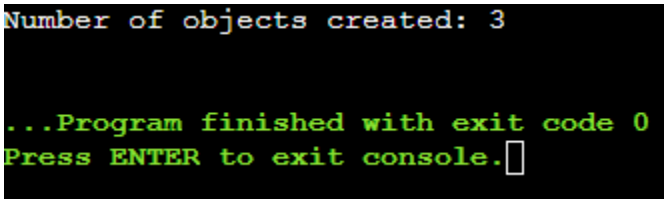
33.) Program to create a simple class to find out the area and perimeter of rectangle using super and this keyword.

Ans.)
```
class Shape {
    protected double length;
    protected double width;
    public Shape(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public void displayDimensions() {
        System.out.println("Length: " + this.length + ", Width: " + this.width);
    }
}
class Rectangle extends Shape {
    public Rectangle(double length, double width) {
        super(length, width); // Calls the constructor of the parent class
    }
    public double calculateArea() {
        return this.length * this.width; // Refers to the current object's variables
    }
    public double calculatePerimeter() {
        return 2 * (this.length + this.width); // Refers to the current object's variables
    }
    public void displayDetails() {
        super.displayDimensions(); // Calls the method from the parent class
        System.out.println("Area: " + calculateArea());
        System.out.println("Perimeter: " + calculatePerimeter());
    }
}
public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(10.5, 5.0);
        rectangle.displayDetails();
    }
}
```



34.) Program to count the number of objects created for class using static member functions.

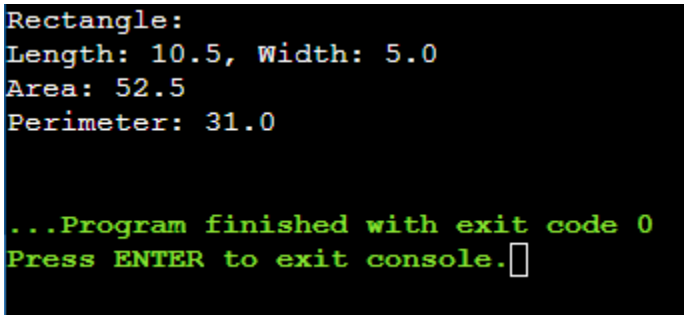
Ans.)
```
public class ObjectCounter {
    private static int count = 0;

    public ObjectCounter() {
        count++;
    }

    public static int getObjectCount() {
        return count;
    }

    public static void main(String[] args) {
        ObjectCounter obj1 = new ObjectCounter();
        ObjectCounter obj2 = new ObjectCounter();
        ObjectCounter obj3 = new ObjectCounter();

        System.out.println("Number of objects created: " + ObjectCounter.getObjectCount());
    }
}
```



35.) Program to design a class using abstract methods and abstract classes.

Ans.)
```
abstract class Shape {
    public abstract double calculateArea();
    public abstract double calculatePerimeter();
    public void displayDetails() {
        System.out.println("This is a shape.");
    }
}
class Rectangle extends Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    @Override
    public double calculateArea() {
        return length * width;
    }
    @Override
```

```
    public double calculatePerimeter() {
        return 2 * (length + width);
    }
    @Override
    public void displayDetails() {
        System.out.println("Rectangle:");
        System.out.println("Length: " + length + ", Width: " + width);
        System.out.println("Area: " + calculateArea());
        System.out.println("Perimeter: " + calculatePerimeter());
    }
}
public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(10.5, 5.0);
        rectangle.displayDetails();
    }
}
```



36.) Program to demonstrate the use of multilevel inheritance.

Ans.)
```
class Animal {
    public void eat() {
        System.out.println("This animal eats food.");
    }
}
class Mammal extends Animal {
    public void walk() {
        System.out.println("This mammal walks on land.");
    }
}
class Dog extends Mammal {
    public void bark() {
        System.out.println("The dog barks.");
    }
}
public class MultilevelInheritance {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.eat();
        dog.walk();
```

```
        dog.bark();
    }
}
```

```
This animal eats food.
This mammal walks on land.
The dog barks.


...Program finished with exit code 0
Press ENTER to exit console.
```

37.) Program to demonstrate the use of multiple inheritance.

Ans.)
```
interface Animal {
    void eat();
}
interface Bird {
    void fly();
}
class Bat implements Animal, Bird {
    @Override
    public void eat() {
        System.out.println("Bat eats insects.");
    }
    @Override
    public void fly() {
        System.out.println("Bat flies in the night.");
    }
}
public class MultipleInheritance{
    public static void main(String[] args) {
        Bat bat = new Bat();
        bat.eat();
        bat.fly();
    }
}
```

```
Bat eats insects.
Bat flies in the night.


...Program finished with exit code 0
Press ENTER to exit console.
```