MANIT

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

Academic Year: 2025-26

Department: Mathematics, Bioinformatics & Computer Application (MBC)

Name of Assignment: Programming LAB In JAVA

Date of Submission: 10/11/2025

Full Name: Shivam Verma **Subject Coord.: Dr. Ghanshyam Singh Thakur**

Scholar No.: 24204031116
Course: MCA (IIIrd sem.)
Subject Code : CA24306

Student Sign: **Professor Sign:**

===== Combined Java Code from All Folders =====

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a1.java =====
//Write a Java program to check whether a number is even or odd using if-else.

import java.util.*;

public class a1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = sc.nextInt();
        if (n % 2 == 0) {
            System.out.println(n + " is even.");
        } else {
            System.out.println(n + " is odd.");    //   ((n & 1) == 1)
        }
        sc.close();
    }
}




===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a10.java =====
public class a10 {
    public static void main(String[] args) {
        // Check if any arguments are provided
        if (args.length == 0) {
            System.out.println("No command-line arguments provided.");
            return;
        }

        System.out.println("Arguments in reverse order:");
        for (int i = args.length - 1; i >= 0; i--) {
            System.out.println(args[i]);
        }
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a2.java =====

```
// Create a program to take a number from the command line and display its multiplication table using a for loop.

import java.util.Scanner;

public class a2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("enter a number : ");
        int n = sc.nextInt();
        System.out.println("Table of " + n + "is : ");
        for (int i=1; i<=10; i++){
            System.out.println( n + "*" + i + "= " + n*i);
        }
        sc.close();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a3.java =====

```
// Write a Java program to find the largest of three numbers using nested if-else

import java.util.*;
public class a3 {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("enter a first number :  ");
    int n1 = sc.nextInt();
    System.out.print("enter a second number : ");
    int n2 = sc.nextInt();
    System.out.print("enter a third number : ");
    int n3 = sc.nextInt();

        if (n1>n2 && n1>n3){
            System.out.println("Largest number is : " + n1);
        }else if(n2>n3){
            System.out.println("largest number is : " + n2);
        }else {
            System.out.println("largest number is : " + n3);
        }
    sc.close();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a4.java =====
//Develop a program to reverse a given number using a while loop.

import java.util.*;

public class a4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        int rev = 0;
        while (n != 0) {
            int digit = n % 10;
            rev = rev * 10 + digit;
            n = n / 10;
        }

        System.out.println("Reversed number: " + rev);
        sc.close();
    }
}



===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a5.java =====
import java.util.*;
```

```
public class a5 {
    public static void main(String[] args) {
        Scanner golu = new Scanner(System.in);
        System.out.print("Enter the size of array: ");
        int n = golu.nextInt();

        int a[] = new int[n];

        System.out.println("Enter " + n + " integers:");
        for (int i = 0; i < n; i++) {
            a[i] = golu.nextInt();
        }

        System.out.println("Even numbers in the array:");
        for (int i = 0; i < n; i++) {
            if (a[i] % 2 == 0) {
                System.out.println(a[i]);
            }
        }

        golu.close();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a6.java =====

```
//Implement a program to sort an array of integers in ascending order using a loop.
```

```
public class a6 {
    public static void main(String[] args) {
        int a[] = {34, 45, 45, 22, 32, 34, 45, 65};
        int n = a.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - 1; j++) {
                if (a[j] > a[j + 1]) {
                    int temp = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = temp;
                }
            }
        }
        System.out.println("Sorted array in ascending order:");
        for (int i = 0; i < n; i++) {
            System.out.print(a[i] + " ");
        }
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a7.java =====

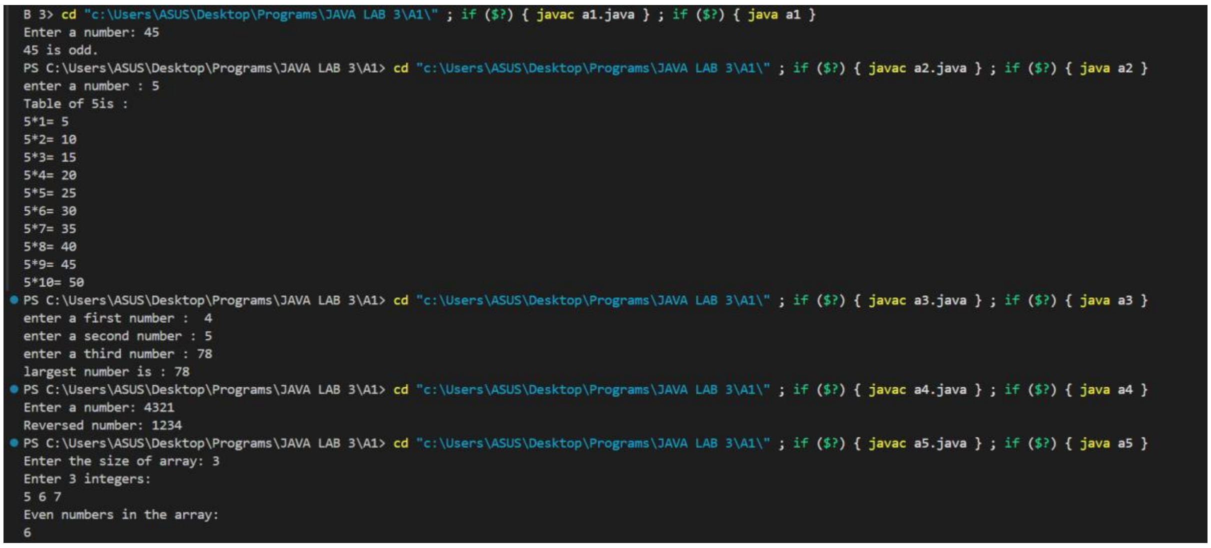
```
// Write a program to count the number of vowels, consonants, digits, and special characters in a given string.
import java.util.*;
public class a7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String s = sc.nextLine();

        int vowels = 0, consonants = 0, digits = 0, special = 0;

        for (int i = 0; i < s.length(); i++) {
            char ch = s.charAt(i);

            if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
                    ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
                    vowels++;
                } else {
                    consonants++;
                }
            } else if (ch >= '0' && ch <= '9') {
                digits++;
            } else if (ch != ' ') {
                special++;
            }
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
        System.out.println("Digits: " + digits);
        System.out.println("Special Characters: " + special);

        sc.close();
    }
}
```

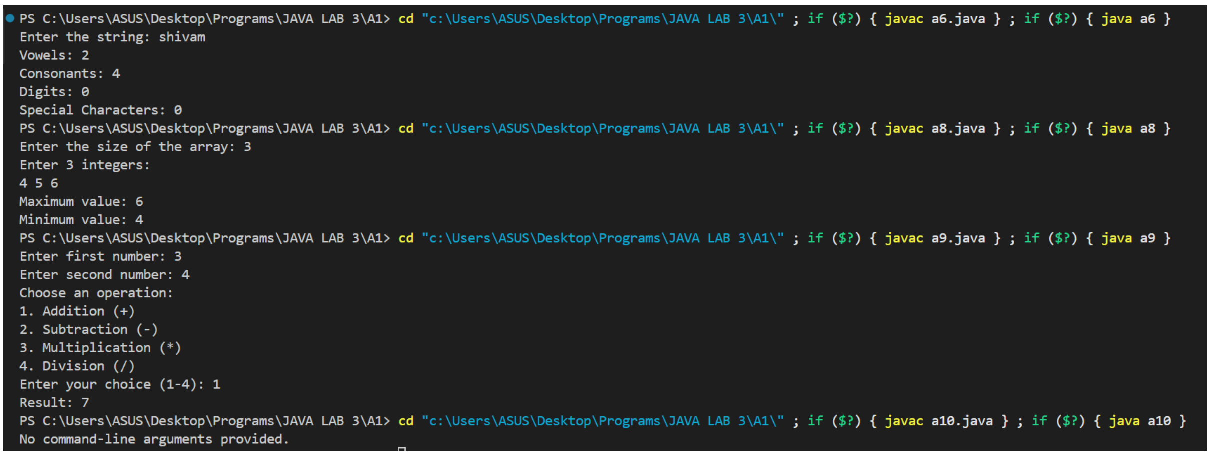
```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a8.java =====
// Accept an integer array and find the maximum and minimum values.
import java.util.*;
public class a8 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = sc.nextInt();
        int a[] = new int[n];
        System.out.println("Enter " + n + " integers:");
        for (int i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }
        int max= a[0];
        int min= a[0];

        // Find max and min
        for (int i = 1; i < n; i++) {
            if (a[i] > max)
                max = a[i];
            if (a[i] < min)
                min = a[i];
        }
        System.out.println("Maximum value: " + max);
        System.out.println("Minimum value: " + min);

        sc.close();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A1\a9.java =====

```
// Write a program using a switch statement to create a simple calculator.
import java.util.*;

public class a9 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        int num1 = sc.nextInt();
        System.out.print("Enter second number: ");
        int num2 = sc.nextInt();

        System.out.println("Choose an operation:");
        System.out.println("1. Addition (+)");
        System.out.println("2. Subtraction (-)");
        System.out.println("3. Multiplication (*)");
        System.out.println("4. Division (/)");
        System.out.print("Enter your choice (1-4): ");
        int choice = sc.nextInt();

        int result;

        switch (choice) {
            case 1:
```

```
            result = num1 + num2;
            System.out.println("Result: " + result);
            break;
          case 2:
            result = num1 - num2;
            System.out.println("Result: " + result);
            break;
          case 3:
            result = num1 * num2;
            System.out.println("Result: " + result);
            break;
          case 4:
            if (num2 != 0) {
              result = num1 / num2;
              System.out.println("Result: " + result);
            } else {
              System.out.println("Error: Cannot divide by zero!");
            }
            break;
          default:
            System.out.println("Invalid choice! Please select from 1 to 4.");
        }

        sc.close();
    }
}
```


```
B 3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a1.java } ; if ($?) { java a1 }
Enter a number: 45
45 is odd.
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a2.java } ; if ($?) { java a2 }
enter a number : 5
Table of 5is :
5*1= 5
5*2= 10
5*3= 15
5*4= 20
5*5= 25
5*6= 30
5*7= 35
5*8= 40
5*9= 45
5*10= 50
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a3.java } ; if ($?) { java a3 }
enter a first number :  4
enter a second number : 5
enter a third number : 78
largest number is : 78
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a4.java } ; if ($?) { java a4 }
Enter a number: 4321
Reversed number: 1234
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a5.java } ; if ($?) { java a5 }
Enter the size of array: 3
Enter 3 integers:
5 6 7
Even numbers in the array:
6
```


```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a6.java } ; if ($?) { java a6 }
Enter the string: shivam
Vowels: 2
Consonants: 4
Digits: 0
Special Characters: 0
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a8.java } ; if ($?) { java a8 }
Enter the size of the array: 3
Enter 3 integers:
4 5 6
Maximum value: 6
Minimum value: 4
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a9.java } ; if ($?) { java a9 }
Enter first number: 3
Enter second number: 4
Choose an operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter your choice (1-4): 1
Result: 7
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A1\" ; if ($?) { javac a10.java } ; if ($?) { java a10 }
No command-line arguments provided.
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a11.java =====

```java
//Q11. Define a class Employee with attributes id, name, salary and create multiple objects to display details.
class a11 {
    int id;
    String name;
    double salary;

    a11(int i, String n, double s) {
        id = i; name = n; salary = s;
    }

    void display() {
        System.out.println(id + " " + name + " " + salary);
    }

    public static void main(String[] args) {
        a11 e1 = new a11(1, "Shivam", 50000);
        a11 e2 = new a11(2, "Anuj", 60000);
        e1.display();
        e2.display();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a12.java =====

```java
// Q12. Create a class with parameterized, default, and copy constructors.
class a12 {
    int x;
    a12() {
        x = 0;
    }
    a12(int a) {
         x = a;
        }
    a12(a12 s) {
         x = s.x;
        }

    void show() { System.out.println("x = " + x); }

    public static void main(String[] args) {
        a12 s1 = new a12();
        a12 s2 = new a12(10);
        a12 s3 = new a12(s2);
        s1.show(); s2.show(); s3.show();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a13.java =====

```
// Q13. Implement a class Student that contains a constructor to initialize name, roll number, and grade.
class a13 {
    String name;
    int roll;
    char grade;
    a13(String n, int r, char g) {
        name = n; roll = r; grade = g;
    }

    void display() {
        System.out.println(name + " " + roll + " " + grade);
    }

    public static void main(String[] args) {
        a13 s = new a13("Shivam", 101, 'A');
        s.display();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a14.java =====
// Q14. Demonstrate the use of static variables and methods inside a class.
class a14 {
    static int count = 0;

    a14() {
        count++;
    }

    static void showCount() {
        System.out.println("Objects created: " + count);
    }
    public static void main(String[] args) {
        new a14(); new a14(); new a14();
        a14.showCount();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a15.java =====

```
// Q15. Create an abstract class Shape with an abstract method area() and concrete method display(). Extend it in Circle and Rectangle.
abstract class Shape {
    abstract void area();
    void display() { System.out.println("This is a shape."); }
}

class Circle extends Shape {
    int r = 5;
    void area() { System.out.println("Area = " + (3.14 * r * r)); }
}

class Rectangle extends Shape {
    int l = 4, b = 3;
    void area() { System.out.println("Area = " + (l * b)); }
}

class a15 {
    public static void main(String[] args) {
        Shape s1 = new Circle();
        Shape s2 = new Rectangle();
        s1.display(); s1.area();
        s2.display(); s2.area();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a16.java =====
// Q16. Show the use of this and super keywords with constructors in base and derived classes.
class Base {
    Base() { System.out.println("Base class constructor"); }
}

class Derived extends Base {
    Derived() {
        super();
        System.out.println("Derived class constructor using super()");
    }
}

class a16 {
    int x;
    a16(int x) {
        this.x = x;
        System.out.println("this.x = " + x);
    }

    public static void main(String[] args) {
        new Derived();
        new a16(10);
    }
}
```

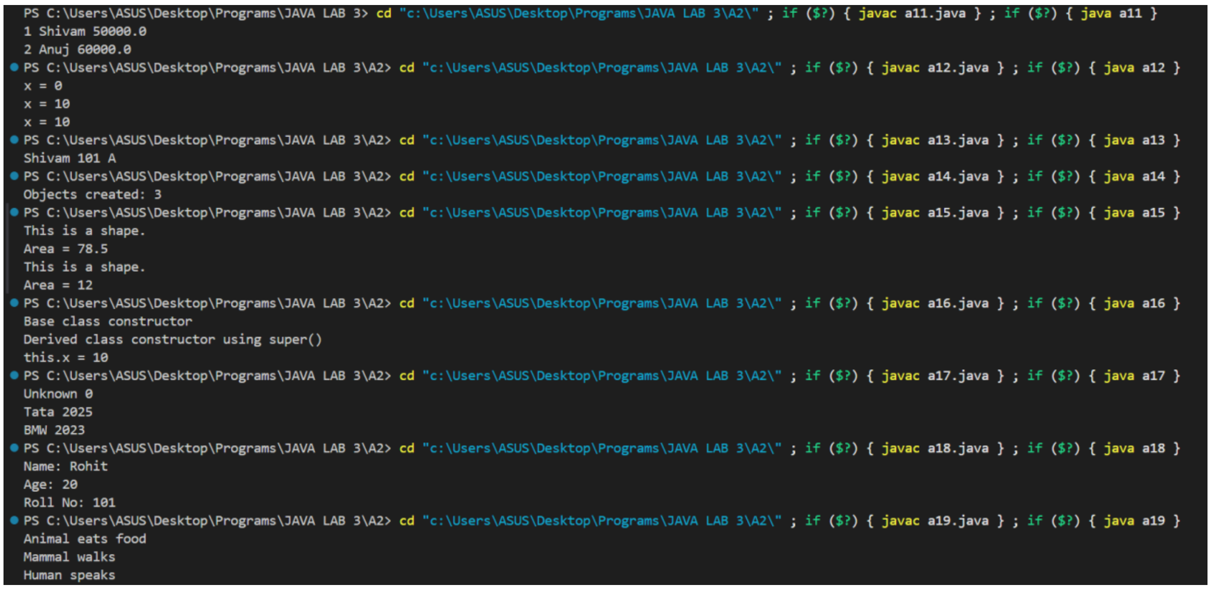
```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a17.java =====
// Q17. Create a class Car with constructor overloading and method to display vehicle information.
class a17 {
    String model;
    int year;

    a17() { model = "Unknown"; year = 0; }
    a17(String m) { model = m; year = 2025; }
    a17(String m, int y) { model = m; year = y; }

    void display() { System.out.println(model + " " + year); }

    public static void main(String[] args) {
        new a17().display();
        new a17("Tata").display();
        new a17("BMW", 2023).display();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a18.java =====
// Q18. Write a program to implement single inheritance using Person â†’ Student.
class Person {
    String name;
    int age;

    void input(String n, int a) {
        name = n;
        age = a;
    }

    void show() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

class Student extends Person {
    int roll;

    void setRoll(int r) {
        roll = r;
    }

    void display() {
        show();
        System.out.println("Roll No: " + roll);
    }
}

class a18 {
    public static void main(String[] args) {
        Student s = new Student();
        s.input("Rohit", 20);
        s.setRoll(101);
        s.display();
    }
}
```

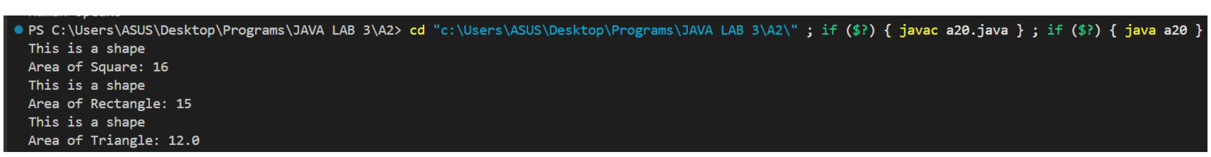
```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a19.java =====
// Q19. Demonstrate multilevel inheritance using classes Animal â†’ Mammal â†’ Human.
class Animal {
    void eat() {
        System.out.println("Animal eats food");
    }
}

class Mammal extends Animal {
    void walk() {
        System.out.println("Mammal walks");
    }
}

class Human extends Mammal {
    void speak() {
        System.out.println("Human speaks");
    }
}

class a19 {
    public static void main(String[] args) {
        Human h = new Human();
        h.eat();

        h.walk();

        h.speak();

    }

}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A2\a20.java =====

```
// Q20. Implement hierarchical inheritance where a base class Shape has subclasses Square,
Rectangle, and Triangle.

class Shape {

    void show() {

        System.out.println("This is a shape");

    }

}


class Square extends Shape {

    void area(int side) {

        System.out.println("Area of Square: " + (side * side));

    }

}


class Rectangle extends Shape {

    void area(int l, int b) {

        System.out.println("Area of Rectangle: " + (l * b));

    }

}


class Triangle extends Shape {
    void area(double b, double h) {
        System.out.println("Area of Triangle: " + (0.5 * b * h));
    }
}

class a20 {
    public static void main(String[] args) {
        Square s = new Square();
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();

        s.show();
        s.area(4);

        r.show();
        r.area(5, 3);

        t.show();
        t.area(6, 4);
    }
}
```



```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a11.java } ; if ($?) { java a11 }
1 Shivam 50000.0
2 Anuj 60000.0
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a12.java } ; if ($?) { java a12 }
x = 0
x = 10
x = 10
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a13.java } ; if ($?) { java a13 }
Shivam 101 A
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a14.java } ; if ($?) { java a14 }
Objects created: 3
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a15.java } ; if ($?) { java a15 }
This is a shape.
Area = 78.5
This is a shape.
Area = 12
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a16.java } ; if ($?) { java a16 }
Base class constructor
Derived class constructor using super()
this.x = 10
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a17.java } ; if ($?) { java a17 }
Unknown 0
Tata 2025
BMW 2023
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a18.java } ; if ($?) { java a18 }
Name: Rohit
Age: 20
Roll No: 101
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a19.java } ; if ($?) { java a19 }
Animal eats food
Mammal walks
Human speaks
```



```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A2\" ; if ($?) { javac a20.java } ; if ($?) { java a20 }
This is a shape
Area of Square: 16
This is a shape
Area of Rectangle: 15
This is a shape
Area of Triangle: 12.0
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a21.java =====

```
// Q21. Show how constructor chaining works in multilevel inheritance.
class A {
    A() {
        System.out.println("Constructor of A");
    }
}

class B extends A {
    B() {
        super();
        System.out.println("Constructor of B");
    }
}

class C extends B {
    C() {
        super();
        System.out.println("Constructor of C");
    }
}

class a21 {
    public static void main(String[] args) {
        new C();
    }
}




===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a22.java =====
// Q22. Explain and demonstrate the limitation of multiple inheritance in Java using classes.
class A {
    void show() {
        System.out.println("Hello from A");
    }
}

class B {
    void show() {
        System.out.println("Hello from B");
    }
}

// This will cause an error in Java
// class C extends A, B { } // Not allowed - multiple inheritance with classes

// Java supports multiple inheritance through interfaces
interface X {
    void display();
}

interface Y {
    void display();
}
```

```
class C implements X, Y {
    public void display() {
        System.out.println("Multiple inheritance using interfaces works fine");
    }
}

class a22 {
    public static void main(String[] args) {
        System.out.println("Java does NOT allow multiple inheritance using classes.");
        System.out.println("But it allows it using interfaces.\n");

        C obj = new C();
        obj.display();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a23.java =====
// Q23. Write a program to show hybrid inheritance using interface and classes.
interface A {
    void show();
}

class B {
    void display() {
        System.out.println("Class B method");
    }
}
```

```
class C extends B implements A {
    public void show() {
        System.out.println("Interface A method");
    }
}

class a23 {
    public static void main(String[] args) {
        C obj = new C();
        obj.display();
        obj.show();
    }
}




===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a24.java =====
// Q24. Write a program to show compile-time polymorphism (method overloading).
class MathOperation {
    void add(int a, int b) {
        System.out.println("Sum = " + (a + b));
    }

    void add(int a, int b, int c) {
        System.out.println("Sum = " + (a + b + c));
    }
}

class a24 {
    public static void main(String[] args) {
        MathOperation m = new MathOperation();
        m.add(5, 10);
        m.add(2, 3, 4);
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a25.java =====

// Q25. Write a program to demonstrate runtime polymorphism using Animal and its subclasses with method overriding.

```
class Animal {
    void sound() {
        System.out.println("Animal makes sound");
    }
}

class Dog extends Animal {
    void sound() {
        System.out.println("Dog barks");
    }
}

class Cat extends Animal {
    void sound() {
        System.out.println("Cat meows");
    }
}
```

```
class a25 {
    public static void main(String[] args) {
        Animal a;
        a = new Dog();
        a.sound();
        a = new Cat();
        a.sound();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a26.java =====

```
// Q26. Show how method resolution occurs in case of overridden methods and dynamic method dispatch.
class Parent {
    void msg() {
        System.out.println("Message from Parent");
    }
}

class Child extends Parent {
    void msg() {
        System.out.println("Message from Child");
    }
}

class a26 {
    public static void main(String[] args) {
        Parent p = new Child(); // reference of parent, object of child
        p.msg(); // calls child method (runtime binding)
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a27.java =====

```java
// Q27. Implement a scenario where polymorphism is used to calculate the salary of different types of employees.
class Employee {
    void salary() {
        System.out.println("Employee salary");
    }
}

class Manager extends Employee {
    void salary() {
        System.out.println("Manager salary = 80000");
    }
}

class Clerk extends Employee {
    void salary() {
        System.out.println("Clerk salary = 30000");
    }
}

class Salesperson extends Employee {
    void salary() {
        System.out.println("Salesperson salary = 40000");
    }
}

class a27 {
    public static void main(String[] args) {
        Employee e;
        e = new Manager();
        e.salary();
        e = new Clerk();
        e.salary();
        e = new Salesperson();
        e.salary();
    }
}
```

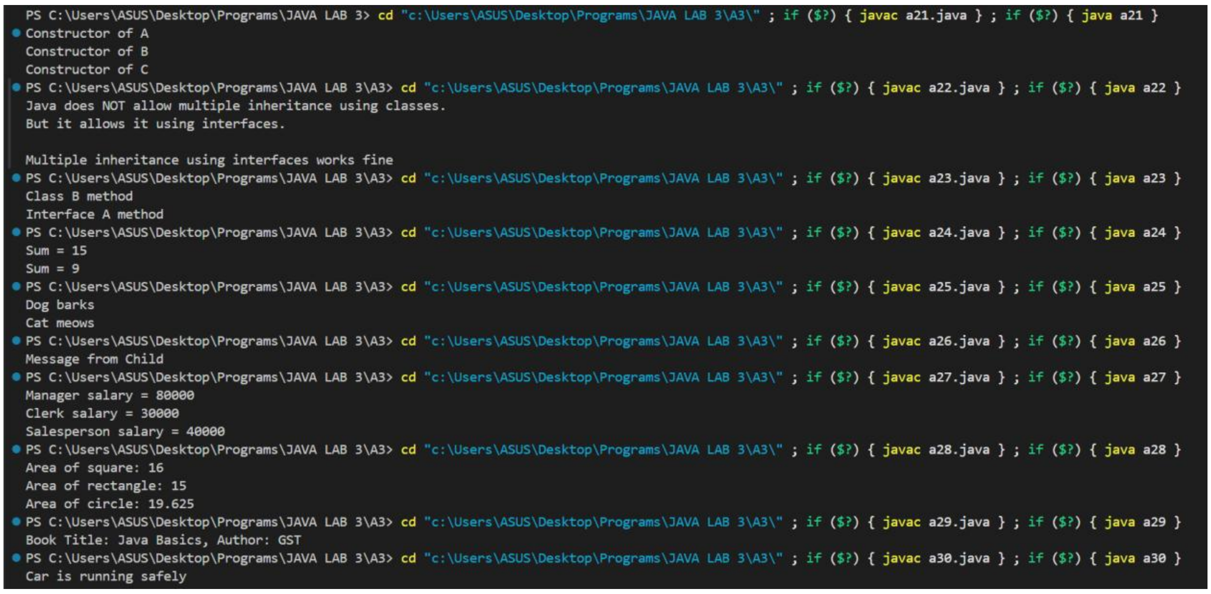
===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a28.java =====

```
// Q28. Create a class with multiple versions of calculateArea() (for circle, square, rectangle).
class Area {
    void calculateArea(int side) {
        System.out.println("Area of square: " + (side * side));
    }

    void calculateArea(int l, int b) {
        System.out.println("Area of rectangle: " + (l * b));
    }

    void calculateArea(double r) {
        System.out.println("Area of circle: " + (3.14 * r * r));
    }
}

class a28 {
    public static void main(String[] args) {
        Area a = new Area();
        a.calculateArea(4);
        a.calculateArea(5, 3);
        a.calculateArea(2.5);
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a29.java =====
// Q29. Override the toString() method in a user-defined class Book.
class Book {
    String title;
    String author;

    Book(String t, String a) {
        title = t;
        author = a;
    }

    public String toString() {
        return "Book Title: " + title + ", Author: " + author;
    }
}

class a29 {
    public static void main(String[] args) {
        Book b = new Book("Java Basics", "GST");
        System.out.println(b);
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A3\a30.java =====

```
// Q30. Show how method overriding works when superclass reference is used to call a subclass method.
class Vehicle {
    void run() {
        System.out.println("Vehicle is running");
    }
}

class Car extends Vehicle {
    void run() {
        System.out.println("Car is running safely");
    }
}

class a30 {
    public static void main(String[] args) {
        Vehicle v = new Car();
        v.run();
    }
}
```



```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a21.java } ; if ($?) { java a21 }
Constructor of A
Constructor of B
Constructor of C
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a22.java } ; if ($?) { java a22 }
Java does NOT allow multiple inheritance using classes.
But it allows it using interfaces.

Multiple inheritance using interfaces works fine
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a23.java } ; if ($?) { java a23 }
Class B method
Interface A method
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a24.java } ; if ($?) { java a24 }
Sum = 15
Sum = 9
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a25.java } ; if ($?) { java a25 }
Dog barks
Cat meows
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a26.java } ; if ($?) { java a26 }
Message from Child
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a27.java } ; if ($?) { java a27 }
Manager salary = 80000
Clerk salary = 30000
Salesperson salary = 40000
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a28.java } ; if ($?) { java a28 }
Area of square: 16
Area of rectangle: 15
Area of circle: 19.625
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a29.java } ; if ($?) { java a29 }
Book Title: Java Basics, Author: GST
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A3\" ; if ($?) { javac a30.java } ; if ($?) { java a30 }
Car is running safely
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\pack1\A.java =====

```
// Q37. Create two packages: pack1 with class A, and pack2 with class B that imports A and uses its members accordingly.

package pack1;


public class A {

    public void msg() {

        System.out.println("Hello from class A in pack1");

    }

}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\pack2\B.java =====

```
package pack2;

import pack1.A;
```

```
public class B {
   public void show() {
      A obj = new A();
      obj.msg();
      System.out.println("Hello from class B in pack2");
   }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\utility\AccessDemo.java =====
// Q36. Demonstrate use of all access modifiers (private, default, protected, public).
package utility;

public class AccessDemo {
   private int a = 10;
   int b = 20;          // default
   protected int c = 30;
   public int d = 40;

   public void show() {
      System.out.println("private: " + a);
      System.out.println("default: " + b);
      System.out.println("protected: " + c);
      System.out.println("public: " + d);
   }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\utility\MathTools.java =====
// Q35. Create a user-defined package utility with a class MathTools that has methods for factorial
and prime check.
package utility;


public class MathTools {
    public int factorial(int n) {
        int fact = 1;
        for (int i = 1; i <= n; i++)
            fact *= i;
        return fact;
    }


    public boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i <= n / 2; i++)
            if (n % i == 0) return false;
        return true;
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a31.java =====
// Q31. Demonstrate the difference between method overloading and overriding in a single program.
class MathOp {
    // Method Overloading (same name, different parameters)
    void add(int a, int b) {
        System.out.println("Overloading: Sum = " + (a + b));
    }

    void add(double a, double b) {
        System.out.println("Overloading: Sum = " + (a + b));
    }
}

class AdvMath extends MathOp {
    // Method Overriding (same name, same parameters)
    void add(int a, int b) {
        System.out.println("Overriding: Sum = " + (a + b + 10));
    }
}

class a31 {
    public static void main(String[] args) {
        AdvMath obj = new AdvMath();

        obj.add(5, 5);      // overriding
        obj.add(2.5, 3.5);  // overloading
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a32.java =====
// Q32. Define an interface Printable and implement it in classes Book and Magazine.
interface Printable {
    void print();
}

class Book implements Printable {
    public void print() {
        System.out.println("Printing Book details...");
    }
}

class Magazine implements Printable {
    public void print() {
        System.out.println("Printing Magazine details...");
    }
}

class a32 {
    public static void main(String[] args) {
        Printable p1 = new Book();
        Printable p2 = new Magazine();

        p1.print();
        p2.print();
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a33.java =====

```
// Q33. Create an interface Bank with method getRateOfInterest() and implement it in SBI, ICICI, and Axis.
interface Bank {
    int getRateOfInterest();
}

class SBI implements Bank {
    public int getRateOfInterest() { return 7; }
}

class ICICI implements Bank {
    public int getRateOfInterest() { return 8; }
}

class Axis implements Bank {
    public int getRateOfInterest() { return 9; }
}

class a33 {
    public static void main(String[] args) {
        Bank b;

        b = new SBI();
        System.out.println("SBI ROI: " + b.getRateOfInterest() + "%");

        b = new ICICI();
        System.out.println("ICICI ROI: " + b.getRateOfInterest() + "%");

        b = new Axis();
        System.out.println("Axis ROI: " + b.getRateOfInterest() + "%");
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a34.java =====

```
// Q34. Write a program to show how multiple inheritance is achieved using interfaces.
interface A {
    void showA();
}

interface B {
    void showB();
}

class C implements A, B {
    public void showA() {
        System.out.println("Method from Interface A");
    }

    public void showB() {
        System.out.println("Method from Interface B");
    }
}

class a34 {
    public static void main(String[] args) {
        C obj = new C();
        obj.showA();
        obj.showB();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a35.java =====
// Q35. Using MathTools from user-defined package.
import utility.MathTools;

class a35 {
    public static void main(String[] args) {
        MathTools m = new MathTools();
        System.out.println("Factorial of 5: " + m.factorial(5));
        System.out.println("Is 7 prime? " + m.isPrime(7));
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a36.java =====
// Q36. Accessing variables with different access modifiers.
import utility.AccessDemo;

class a36 extends AccessDemo {
    public static void main(String[] args) {
        AccessDemo obj = new AccessDemo();
        obj.show();

        // Access from subclass
        a36 sub = new a36();
        System.out.println("\nAccessing from subclass:");
        // System.out.println(sub.a); // private - not accessible
        // System.out.println(sub.b); // default - not accessible outside package
        System.out.println("protected: " + sub.c);

        System.out.println("public: " + sub.d);

    }

}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a37.java =====

// Q37. Program to show use of two packages (pack1 and pack2).

import pack2.B;


class a37 {

    public static void main(String[] args) {

        B b = new B();

        b.show();

    }

}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a38.java =====

// Q38. Write a program to create a thread by extending the Thread class and display current thread
info.

class MyThread extends Thread {

    public void run() {

        System.out.println("Thread is running...");

        System.out.println("Current Thread: " + Thread.currentThread());

    }

}
```

```
class a38 {
    public static void main(String[] args) {
        MyThread t = new MyThread();
        t.start();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a39.java =====
// Q39. Write a program to implement a thread using the Runnable interface.
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("Thread running using Runnable interface");
    }
}

class a39 {
    public static void main(String[] args) {
        Thread t = new Thread(new MyRunnable());
        t.start();
    }
}
```

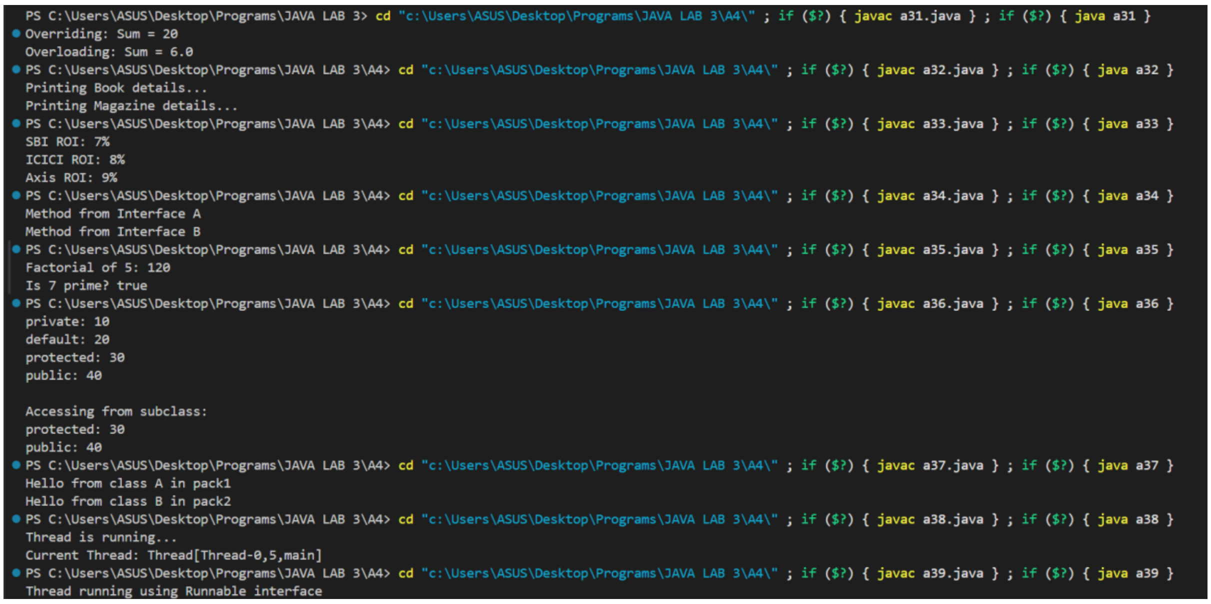
```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a40.java =====
// Q40. Create two threads: one to print even numbers and another to print odd numbers up to 50.
```

```
class EvenThread extends Thread {
    public void run() {
        for (int i = 2; i <= 50; i += 2) {
            System.out.println("Even: " + i);
        }
    }
}

class OddThread extends Thread {
    public void run() {
        for (int i = 1; i <= 50; i += 2) {
            System.out.println("Odd: " + i);
        }
    }
}

class a40 {
    public static void main(String[] args) {
        EvenThread e = new EvenThread();
        OddThread o = new OddThread();
        e.start();
        o.start();
    }
}
```

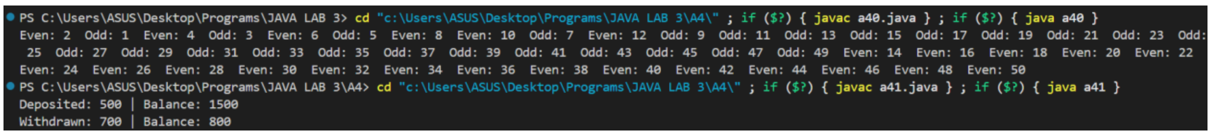
===== FILE: C:\Users\ASUS\Desktop\javaLab\A4\a41.java =====

```
// Q41. Create a program to demonstrate thread synchronization for a banking application (deposit and withdraw).
```

```
class Bank {
    int balance = 1000;

    synchronized void deposit(int amt) {
        balance = balance + amt;
        System.out.println("Deposited: " + amt + " | Balance: " + balance);
    }

    synchronized void withdraw(int amt) {
        if (balance >= amt) {
            balance = balance - amt;
            System.out.println("Withdrawn: " + amt + " | Balance: " + balance);
        } else {
            System.out.println("Not enough balance!");
        }
    }
}

class a41 {
    public static void main(String[] args) {
        Bank b = new Bank();

        Thread t1 = new Thread(() -> b.deposit(500));
        Thread t2 = new Thread(() -> b.withdraw(700));

        t1.start();
        t2.start();
    }
}
```



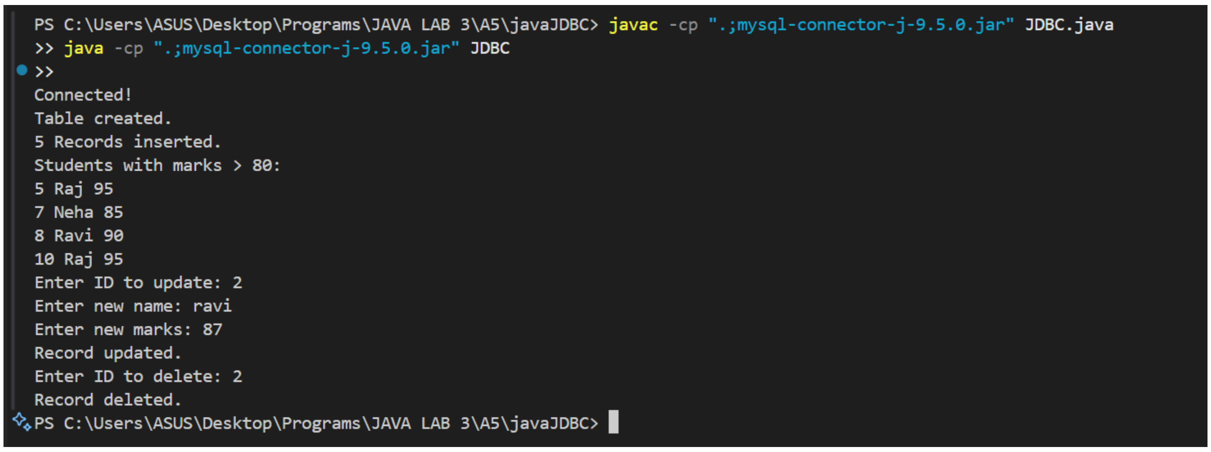
```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a31.java } ; if ($?) { java a31 }
Overriding: Sum = 20
Overloading: Sum = 6.0
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a32.java } ; if ($?) { java a32 }
Printing Book details...
Printing Magazine details...
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a33.java } ; if ($?) { java a33 }
SBI ROI: 7%
ICICI ROI: 8%
Axis ROI: 9%
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a34.java } ; if ($?) { java a34 }
Method from Interface A
Method from Interface B
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a35.java } ; if ($?) { java a35 }
Factorial of 5: 120
Is 7 prime? true
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a36.java } ; if ($?) { java a36 }
private: 10
default: 20
protected: 30
public: 40

Accessing from subclass:
protected: 30
public: 40
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a37.java } ; if ($?) { java a37 }
Hello from class A in pack1
Hello from class B in pack2
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a38.java } ; if ($?) { java a38 }
Thread is running...
Current Thread: Thread[Thread-0,5,main]
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a39.java } ; if ($?) { java a39 }
Thread running using Runnable interface
```



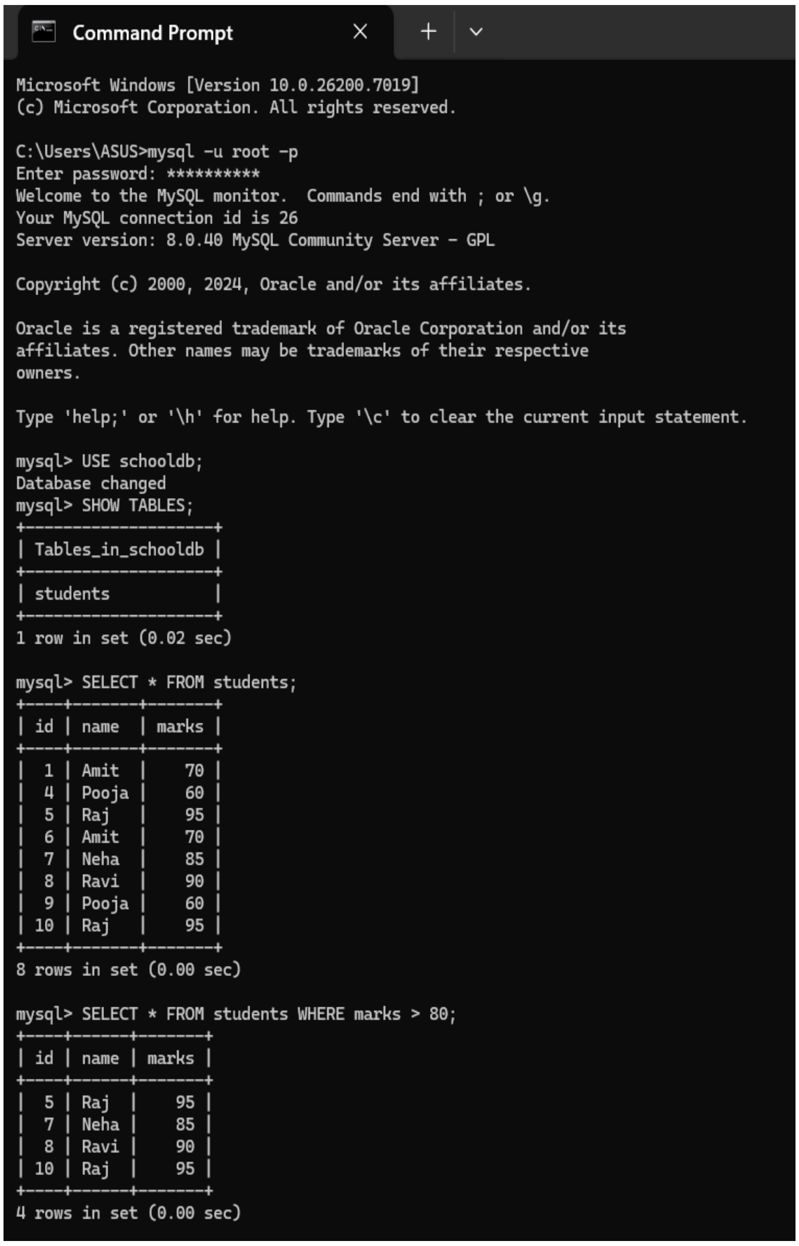
```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a40.java } ; if ($?) { java a40 }
Even: 2  Odd: 1  Even: 4  Odd: 3  Even: 6  Odd: 5  Even: 8  Even: 10  Odd: 7  Even: 12  Odd: 9  Odd: 11  Odd: 13  Odd: 15  Odd: 17  Odd: 19  Odd: 21  Odd: 23  Odd:
 25  Odd: 27  Odd: 29  Odd: 31  Odd: 33  Odd: 35  Odd: 37  Odd: 39  Odd: 41  Odd: 43  Odd: 45  Odd: 47  Odd: 49  Even: 14  Even: 16  Even: 18  Even: 20  Even: 22
Even: 24  Even: 26  Even: 28  Even: 30  Even: 32  Even: 34  Even: 36  Even: 38  Even: 40  Even: 42  Even: 44  Even: 46  Even: 48  Even: 50
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A4\" ; if ($?) { javac a41.java } ; if ($?) { java a41 }
Deposited: 500 | Balance: 1500
Withdrawn: 700 | Balance: 800
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A5\javaJDBC\JDBC.java =====

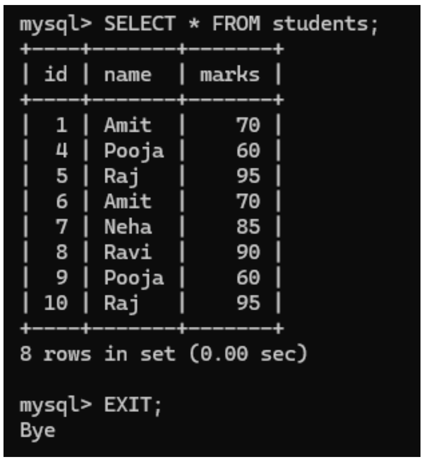
```java
/*Q46. Write a JDBC program to connect with MySQL database and create a table students(id, name, marks).

 Q47. Insert 5 records into the students table using JDBC PreparedStatement.

 Q48. Retrieve and display all student records where marks > 80.

 Q49. Update the name and marks of a student with a specific ID using JDBC.

 Q50. Delete a student record from the table using user input via JDBC and show proper exception handling. */


import java.sql.*;

import java.util.Scanner;


class JDBC {

    public static void main(String[] args) {

        String url  = "jdbc:mysql://localhost:3306/schooldb";
        String user = "root";
        String pass = "";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver"); // load driver once

            try (Connection con = DriverManager.getConnection(url, user, pass);
                 Scanner sc = new Scanner(System.in)) {

                System.out.println("Connected!");

                Statement st = con.createStatement();
                st.execute("CREATE TABLE IF NOT EXISTS students(id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(50), marks INT)");
                System.out.println("Table created.");

                PreparedStatement ps = con.prepareStatement("INSERT INTO students(name, marks)
VALUES(?, ?)");
                String[][] data = {{"Amit","70"},{"Neha","85"},{"Ravi","90"},{"Pooja","60"},{"Raj","95"}};
                for (String[] d : data) {
                    ps.setString(1, d[0]);
                    ps.setInt(2, Integer.parseInt(d[1]));
                    ps.executeUpdate();
                }
                System.out.println("5 Records inserted.");

                ResultSet rs = st.executeQuery("SELECT * FROM students WHERE marks > 80");
                System.out.println("Students with marks > 80:");
                while (rs.next()) {
                    System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getInt(3));
                }
                System.out.print("Enter ID to update: ");
                int id = sc.nextInt();
                sc.nextLine();
                System.out.print("Enter new name: ");
                String name = sc.nextLine();
                System.out.print("Enter new marks: ");
                int marks = sc.nextInt();
                ps = con.prepareStatement("UPDATE students SET name=?, marks=? WHERE id=?");
                ps.setString(1, name);
                ps.setInt(2, marks);
                ps.setInt(3, id);
                ps.executeUpdate();
                System.out.println("Record updated.");

                System.out.print("Enter ID to delete: ");
                int del = sc.nextInt();
                ps = con.prepareStatement("DELETE FROM students WHERE id=?");
                ps.setInt(1, del);
                ps.executeUpdate();
                System.out.println("Record deleted.");
            }

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```



```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5\javaJDBC> javac -cp ".;mysql-connector-j-9.5.0.jar" JDBC.java
>> java -cp ".;mysql-connector-j-9.5.0.jar" JDBC
>>
Connected!
Table created.
5 Records inserted.
Students with marks > 80:
5 Raj 95
7 Neha 85
8 Ravi 90
10 Raj 95
Enter ID to update: 2
Enter new name: ravi
Enter new marks: 87
Record updated.
Enter ID to delete: 2
Record deleted.
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5\javaJDBC>
```



```
Microsoft Windows [Version 10.0.26200.7019]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>mysql -u root -p
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.40 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE schooldb;
Database changed
mysql> SHOW TABLES;
+--------------------+
| Tables_in_schooldb |
+--------------------+
| students           |
+--------------------+
1 row in set (0.02 sec)

mysql> SELECT * FROM students;
+----+-------+-------+
| id | name  | marks |
+----+-------+-------+
|  1 | Amit  |    70 |
|  4 | Pooja |    60 |
|  5 | Raj   |    95 |
|  6 | Amit  |    70 |
|  7 | Neha  |    85 |
|  8 | Ravi  |    90 |
|  9 | Pooja |    60 |
| 10 | Raj   |    95 |
+----+-------+-------+
8 rows in set (0.00 sec)

mysql> SELECT * FROM students WHERE marks > 80;
+----+------+-------+
| id | name | marks |
+----+------+-------+
|  5 | Raj  |    95 |
|  7 | Neha |    85 |
|  8 | Ravi |    90 |
| 10 | Raj  |    95 |
+----+------+-------+
4 rows in set (0.00 sec)
```



```
mysql> SELECT * FROM students;
+----+-------+-------+
| id | name  | marks |
+----+-------+-------+
|  1 | Amit  |    70 |
|  4 | Pooja |    60 |
|  5 | Raj   |    95 |
|  6 | Amit  |    70 |
|  7 | Neha  |    85 |
|  8 | Ravi  |    90 |
|  9 | Pooja |    60 |
| 10 | Raj   |    95 |
+----+-------+-------+
8 rows in set (0.00 sec)

mysql> EXIT;
Bye
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A5\a42.java =====

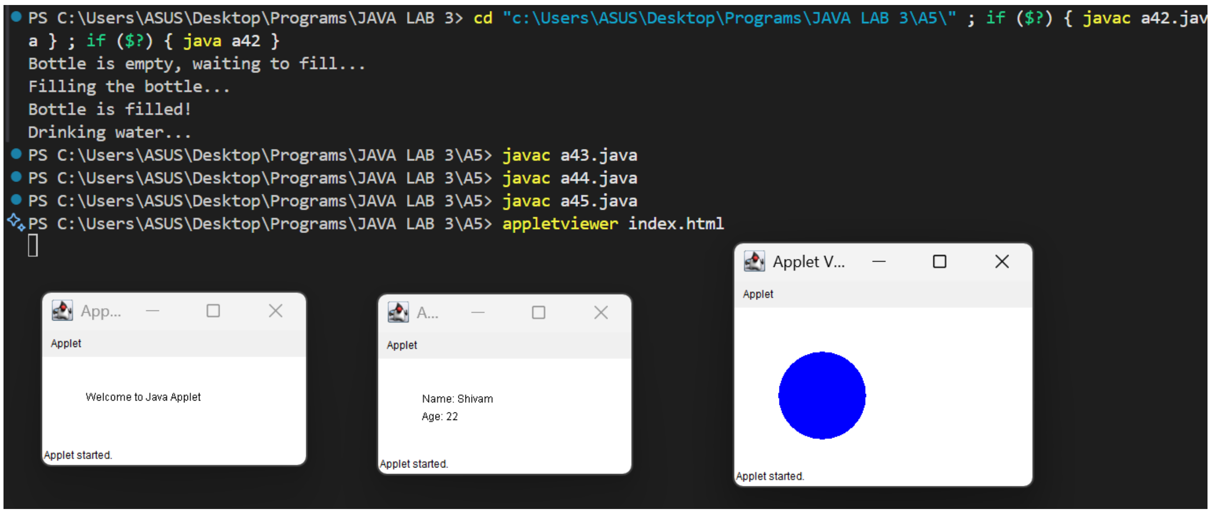
```java
//Q42. Write a program to implement inter-thread communication using wait() and notify() methods.
class Bottle {
    boolean full = false;

    synchronized void drink() {
        if (!full) {
            System.out.println("Bottle is empty, waiting to fill...");
            try { wait(); } catch (Exception e) {}
        }
        System.out.println("Drinking water...");
        full = false;
    }

    synchronized void fill() {
        System.out.println("Filling the bottle...");
        full = true;
        System.out.println("Bottle is filled!");
        notify();
    }
}

class a42 {
    public static void main(String[] args) {
        Bottle b = new Bottle();

        // Drinker thread
        new Thread(() -> b.drink()).start();

        // Small delay before filling
        new Thread(() -> {
            try { Thread.sleep(1000); } catch (Exception e) {}
            b.fill();
        }).start();
    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A5\a43.java =====
// Q43. Simple applet that displays "Welcome to Java Applet".
import java.applet.Applet;
import java.awt.Graphics;
@SuppressWarnings("removal")
public class a43 extends Applet {
    public void paint(Graphics g) {
        g.drawString("Welcome to Java Applet", 50, 50);

    }
}
```

```
===== FILE: C:\Users\ASUS\Desktop\javaLab\A5\a44.java =====
// Q44. Applet that accepts name and age using <param> and displays them.
import java.applet.Applet;
import java.awt.Graphics;
@SuppressWarnings("removal")
public class a44 extends Applet {
    String name;
    int age;

    public void init() {
        name = getParameter("name");
        age = Integer.parseInt(getParameter("age"));
    }

    public void paint(Graphics g) {
        g.drawString("Name: " + name, 50, 50);
        g.drawString("Age: " + age, 50, 70);
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A5\a45.java =====

```
// Q45. Applet that draws and fills a circle with color.
import java.applet.Applet;
import java.awt.*;
@SuppressWarnings("removal")
public class a45 extends Applet {
    public void paint(Graphics g) {
        g.setColor(Color.blue);
        g.fillOval(50, 50, 100, 100);
    }
}
```



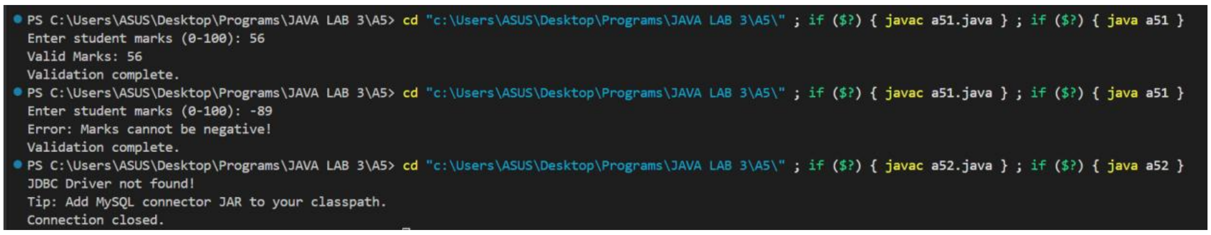
```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5\" ; if ($?) { javac a42.jav
a } ; if ($?) { java a42 }
Bottle is empty, waiting to fill...
Filling the bottle...
Bottle is filled!
Drinking water...
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5> javac a43.java
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5> javac a44.java
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5> javac a45.java
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5> appletviewer index.html
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A5\a51.java =====

```
import java.util.Scanner;

// Custom Exception for Negative Marks
class NegativeMarksException extends Exception {
    public NegativeMarksException(String msg) {
        super(msg);
    }
}

// Custom Exception for Marks > 100
class MarksOutOfRangeException extends Exception {
    public MarksOutOfRangeException(String msg) {
        super(msg);
    }
}

public class a51 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter student marks (0-100): ");
        int marks = sc.nextInt();

        try {
            validateMarks(marks);
            System.out.println("Valid Marks: " + marks);
        }
        catch (NegativeMarksException | MarksOutOfRangeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        finally {
            sc.close();
            System.out.println("Validation complete.");
        }
    }

    static void validateMarks(int marks) throws NegativeMarksException, MarksOutOfRangeException
{
        if (marks < 0)
            throw new NegativeMarksException("Marks cannot be negative!");
        else if (marks > 100)
            throw new MarksOutOfRangeException("Marks cannot exceed 100!");
    }
}
```

===== FILE: C:\Users\ASUS\Desktop\javaLab\A5\a52.java =====

```
import java.sql.*;

public class a52 {
    public static void main(String[] args) {
        Connection con = null;

        try {
            // Load MySQL JDBC Driver
            Class.forName("com.mysql.cj.jdbc.Driver");
            System.out.println("Driver loaded successfully!");

            // Wrong database or credentials to simulate error
            String url = "jdbc:mysql://localhost:3306/fakeDB"; // wrong DB name
            String user = "root";
            String pass = "wrongpass"; // intentionally wrong

            // Try connecting
            con = DriverManager.getConnection(url, user, pass);
            System.out.println("Connected successfully!"); // will not execute

        }
        catch (ClassNotFoundException e) {
            System.out.println("JDBC Driver not found!");
            System.out.println("Tip: Add MySQL connector JAR to your classpath.");
        }
        catch (SQLException e) {
            System.out.println("Database connection failed!");
            System.out.println("Reason: " + e.getMessage());
            System.out.println("SQL State: " + e.getSQLState());
            System.out.println("Error Code: " + e.getErrorCode());
            System.out.println("Tip: Check database name, username, or password.");
        }
        finally {
            try {
                if (con != null) con.close();
                System.out.println("Connection closed.");
            } catch (SQLException e) {
                System.out.println("Error closing connection: " + e.getMessage());
            }
        }
    }
}
```



```
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5\" ; if ($?) { javac a51.java } ; if ($?) { java a51 }
Enter student marks (0-100): 56
Valid Marks: 56
Validation complete.
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5\" ; if ($?) { javac a51.java } ; if ($?) { java a51 }
Enter student marks (0-100): -89
Error: Marks cannot be negative!
Validation complete.
PS C:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5> cd "c:\Users\ASUS\Desktop\Programs\JAVA LAB 3\A5\" ; if ($?) { javac a52.java } ; if ($?) { java a52 }
JDBC Driver not found!
Tip: Add MySQL connector JAR to your classpath.
Connection closed.
```