PR 1

```
import java.util.Scanner;

public class calculator {
   @SuppressWarnings("ConvertToTryWithResources")
   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

      while(true) {

         System.out.print("Enter first number: ");
         double num1 = sc.nextDouble();

         System.out.print("Enter second number: ");
         double num2 = sc.nextDouble();

         System.out.println("\nChoose 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: ");
         int choice = sc.nextInt();

         switch (choice) {
            case 1 ->
            {
                  double result = num1 + num2;
                  System.out.println("Result = " + result);
               }
            case 2 ->
            {
                  double result = num1 - num2;
```

```
            System.out.println("Result = " + result);
        }
      case 3 ->
      {
            double result = num1 * num2;
            System.out.println("Result = " + result);
        }
      case 4 ->
      {
        if(num2 != 0)
        {
            double result = num1 / num2;
            System.out.println("Result = " + result);
        }

        else
        {
            System.out.println("Cannot divide by zero");
        }
      }
      default -> System.out.println("Invalid Choice");
    }

    System.out.println("\nDo you want to continue?");
    System.out.println("1. Yes");
    System.out.println("2. No");

    int ch = sc.nextInt();

    if(ch == 2) {
      System.out.println("Calculator Closed");
      break;
    }
  }
```

```
        sc.close();
    }
}
```

PR 2

```
import java.util.Scanner;

class Product {

    String  name;
    double  price;
    int quantity;

    // Default Constructor
    Product() {
        name = "Unknown Product";
        price = 0;
        quantity = 0;
    }

    // Parameterized Constructor
    Product(String n, double p, int q) {
        name = n;
        price = p;
        quantity = q;
    }

    // Method to calculate total
    double calculateTotal() {
        return price * quantity;
    }
}

public class EcommerceOrder {
    @SuppressWarnings("ConvertToTryWithResources")
```

```
public static void main(String[] args) {

   Scanner sc = new Scanner(System.in);

   // Using default constructor
   @SuppressWarnings("unused")
   Product p1 = new Product();

   // Taking user input
   System.out.print("Enter Product Name: ");
   String name = sc.nextLine();

   System.out.print("Enter Product Price: ");
   double price = sc.nextDouble();

   System.out.print("Enter Quantity: ");
   int quantity = sc.nextInt();

   // Using parameterized constructor
   Product p2 = new Product(name, price, quantity);

   double total = p2.calculateTotal();

   double discount = 0;

   // Discount Policy
   if(total > 5000) {
      discount = total * 0.10;
   }

   double finalAmount = total - discount;

   // Invoice
   System.out.println("\n===== INVOICE =====");
   System.out.println("Product Name : " + p2.name);
```

```
        System.out.println("Price        : " + p2.price);
        System.out.println("Quantity     : " + p2.quantity);
        System.out.println("Total Cost   : " + total);
        System.out.println("Discount     : " + discount);
        System.out.println("Final Amount : " + finalAmount);

        sc.close();
    }
}
```

PR 3.

```
import java.util.Scanner;

class Calculator {

    // Overloaded power methods
    @SuppressWarnings("unused")
    double power(int a, int b) {
        return Math.pow(a, b);
    }

    double power(double a, double b) {
        return Math.pow(a, b);
    }

    // Overloaded absolute methods
    @SuppressWarnings("unused")
    int absolute(int a) {
        return Math.abs(a);
    }

    double absolute(double a) {
        return Math.abs(a);
    }
}
```

```
public class MathOperations {
    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        Calculator obj = new Calculator();

        System.out.println("Choose operation:");
        System.out.println("1. Power");
        System.out.println("2. Absolute");

        int choice = sc.nextInt();

        switch (choice) {
            case 1 -> {
                System.out.print("Enter base: ");
                double base = sc.nextDouble();
                System.out.print("Enter exponent: ");
                double exp = sc.nextDouble();
                System.out.println("Power = " + obj.power(base, exp));
            }
            case 2 -> {
                System.out.print("Enter number: ");
                double num = sc.nextDouble();
                System.out.println("Absolute = " + obj.absolute(num));
            }
            default -> System.out.println("Invalid Choice");
        }

        sc.close();
    }
}
```

PR 4.

```
import java.util.Scanner;

public class ArrayOperations {
   @SuppressWarnings("ConvertToTryWithResources")
   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

      // Input array size
      System.out.print("Enter size of array: ");
      int n = sc.nextInt();

      int arr[] = new int[n];

      // Input elements
      System.out.println("Enter array elements:");

      for(int i = 0; i < n; i++) {
         arr[i] = sc.nextInt();
      }

      // Display elements
      System.out.println("\nArray Elements:");

      for(int i = 0; i < n; i++) {
         System.out.print(arr[i] + " ");
      }

      // Find maximum and minimum
      int max = arr[0];
      int min = arr[0];

      for(int i = 1; i < n; i++) {
```

```
    if(arr[i] > max) {
        max = arr[i];
    }

    if(arr[i] < min) {
        min = arr[i];
    }
}

// Calculate sum
int sum = 0;

for(int i = 0; i < n; i++) {
    sum += arr[i];
}

// Calculate average
double average = (double)sum / n;

// Search element
System.out.print("\nEnter element to search: ");
int key = sc.nextInt();

boolean found = false;

for(int i = 0; i < n; i++) {

    if(arr[i] == key) {
        found = true;
        break;
    }
}

// Display results
System.out.println("\nMaximum Element = " + max);
```

```
        System.out.println("Minimum Element = " + min);
        System.out.println("Sum = " + sum);
        System.out.println("Average = " + average);

        if(found) {
            System.out.println("Element  Found");
        }

        else {
            System.out.println("Element Not Found");
        }

        sc.close();
    }
}
```

PR 5.

```
import java.util.Scanner;

public class HotelBookingSystem {
    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        // 3 floors and 4 rooms per floor
        int rooms[][] = new int[3][4];

        int choice;

        OUTER:
        while (true) {
            System.out.println("\n===== HOTEL BOOKING SYSTEM =====");
```

```
System.out.println("1. View Rooms");
System.out.println("2. Book Room");
System.out.println("3. Exit");
System.out.print("Enter your choice: ");
choice = sc.nextInt();
// View Rooms
switch (choice) {
   case 1 -> {
      System.out.println("\nRoom Status:");
      for(int i = 0; i < 3; i++) {

         System.out.println("Floor " + i + ":");

         for(int j = 0; j < 4; j++) {

            if(rooms[i][j] == 0) {
               System.out.print("[Available]  ");
            }

            else {
               System.out.print("[Booked]  ");
            }
         }

         System.out.println();
      }
   }
   case 2 -> {
      System.out.print("Enter Floor Number (0-2): ");
      int floor = sc.nextInt();
      System.out.print("Enter Room Number (0-3): ");
      int room = sc.nextInt();
      if(rooms[floor][room] == 0) {

         rooms[floor][room] = 1;
```

```
                System.out.println("Room Booked Successfully");
            }

            else {
                System.out.println("Room Already Booked");
            }
        }
        case 3 -> {
            System.out.println("Exiting System...");
            break OUTER;
        }
        default -> System.out.println("Invalid Choice");
      }
    }

    sc.close();
  }
}
```

PR 6.

```
import java.util.Scanner;

// Superclass
class Person {

  String name;
  int age;

  // Method to take input
  void inputDetails() {

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

```
        System.out.print("Enter Name: ");
        name = sc.nextLine();

        System.out.print("Enter Age: ");
        age = sc.nextInt();
    }

    // Method to display details
    void displayDetails() {

        System.out.println("Name = " + name);
        System.out.println("Age = " + age);
    }
}

// Subclass
class Student extends Person {

    int rollNo;

    void inputStudent() {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Roll Number: ");
        rollNo = sc.nextInt();
    }

    void displayStudent() {

        System.out.println("Roll Number = " + rollNo);
    }
}

// Main Class
```

```
public class InheritanceDemo {

    public static void main(String[] args) {

        Student s = new Student();

        // Inherited methods
        s.inputDetails();

        // Student method
        s.inputStudent();

        System.out.println("\n--- Student Details ---");

        // Inherited display method
        s.displayDetails();

        // Student display
        s.displayStudent();
    }
}
```

PR 7.

```
import java.util.Scanner;

// Interface
interface Shape {

    void area();
}

// Circle Class
class Circle implements Shape {

    double radius;
```

```
    Circle(double r) {
        radius = r;
    }

    @SuppressWarnings("override")
    public void area() {

        double result = Math.PI * radius * radius;

        System.out.println("Area of Circle = " + result);
    }
}

// Rectangle Class
class Rectangle implements Shape {

    double length, breadth;

    Rectangle(double l, double b) {
        length = l;
        breadth = b;
    }

    @SuppressWarnings("override")
    public void area() {

        double result = length * breadth;

        System.out.println("Area of Rectangle = " + result);
    }
}

// Main Class
public class InterfaceDemo {
```

```
    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Shape s;

        System.out.println("Choose Shape:");
        System.out.println("1. Circle");
        System.out.println("2. Rectangle");

        int choice = sc.nextInt();

        switch (choice) {
            case 1 -> {
                System.out.print("Enter Radius: ");
                double r = sc.nextDouble();
                s = new Circle(r);
                s.area();
            }
            case 2 -> {
                System.out.print("Enter Length: ");
                double l = sc.nextDouble();
                System.out.print("Enter Breadth: ");
                double b = sc.nextDouble();
                s = new Rectangle(l, b);
                s.area();
            }
            default -> System.out.println("Invalid Choice");
        }

        sc.close();
    }
}
```

PR 8.

```
import java.util.ArrayList;
import java.util.Scanner;

// Abstract Class
abstract class College {

   @SuppressWarnings("unused")
   abstract void calculate();
}

// Derived Class
class Student extends College {

   String name;
   int marks;
   double percentage;
   char grade;
   String result;

   // Constructor
   Student(String n, int m) {
      name = n;
      marks = m;
   }

   // Implement abstract method
   @SuppressWarnings("override")
   void calculate() {

      percentage = (marks / 500.0) * 100;

      if(percentage >= 75) {
         grade = 'A';
         result = "Pass";
```

```
        }

        else if(percentage >= 60) {
            grade = 'B';
            result = "Pass";
        }

        else if(percentage >= 40) {
            grade = 'C';
            result = "Pass";
        }

        else {
            grade = 'F';
            result = "Fail";
        }
    }

    // Display Details
    void display() {

        System.out.println("\nName       : " + name);
        System.out.println("Marks      : " + marks);
        System.out.println("Percentage : " + percentage);
        System.out.println("Grade      : " + grade);
        System.out.println("Result     : " + result);
    }
}

// Main Class
public class CollegeSystem {

    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {
```

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

@SuppressWarnings("Convert2Diamond")
ArrayList<Student> list = new ArrayList<Student>();

System.out.print("Enter Number of Students: ");
int n = sc.nextInt();

sc.nextLine();

// Input Student Details
for(int i = 0; i < n; i++) {

   System.out.println("\nEnter Details of Student " + (i + 1));

   System.out.print("Enter Name: ");
   String name = sc.nextLine();

   System.out.print("Enter Marks out of 500: ");
   int marks = sc.nextInt();

   sc.nextLine();

   Student s = new Student(name, marks);

   s.calculate();

   list.add(s);
}

// Display Details
System.out.println("\n===== STUDENT DETAILS =====");

for(Student s : list) {
   s.display();
```

```
        }

        // Find Topper
        Student topper = list.get(0);

        for(Student s : list) {

            if(s.percentage > topper.percentage) {
                topper = s;
            }
        }

        // Display Topper
        System.out.println("\n===== TOPPER =====");

        System.out.println("Name       : " + topper.name);
        System.out.println("Percentage : " + topper.percentage);

        sc.close();
    }
}
```

PR 9.

```
import java.util.Scanner;

public class ATMApp {

    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        double balance = 5000;

        int choice;
```

```
OUTER:
while (true) {
   System.out.println("\n===== ATM MENU =====");
   System.out.println("1. Check Balance");
   System.out.println("2. Deposit Money");
   System.out.println("3. Withdraw Money");
   System.out.println("4. Exit");
   System.out.print("Enter your choice: ");
   choice = sc.nextInt();
   try {
      // Check Balance
      switch (choice) {
         case 1 -> System.out.println("Current Balance = " + balance);
         case 2 ->                    {
               System.out.print("Enter Deposit Amount: ");
               double amount = sc.nextDouble();
               if(amount <= 0) {
                  throw new IllegalArgumentException(
                        "Invalid Deposit Amount"
                  );
               }     balance += amount;
               System.out.println("Amount Deposited Successfully");
            }
         case 3 ->                    {
               System.out.print("Enter Withdrawal Amount: ");
               double amount = sc.nextDouble();
               if(amount <= 0) {
                  throw new IllegalArgumentException(
                        "Invalid Withdrawal Amount"
                  );
               }     if(amount > balance) {
                  throw new ArithmeticException(
                        "Insufficient Funds"
                  );
```

```
                    }       balance -= amount;
                    System.out.println("Withdrawal Successful");
                }
            case 4 -> {
                System.out.println("Thank You for Using ATM");
                break OUTER;
            }
            default -> System.out.println("Invalid Choice");
        }
      }catch(ArithmeticException | IllegalArgumentException e) {

        System.out.println("Error: " + e.getMessage());
      } finally {
        System.out.println("Transaction Process Completed");
      }
    }

    sc.close();
  }
}
```

PR 10.

```
class ChatUser extends Thread {

  @SuppressWarnings("FieldMayBeFinal")
  private String userName;

  private boolean running = true;
  private boolean paused = false;

  public ChatUser(String name) {
    userName = name;
  }
```

```
// Pause Thread
public synchronized void pauseUser() {
   paused = true;
}

// Resume Thread
public synchronized void resumeUser() {

   paused = false;

   notify();
}

// Stop Thread Safely
public void stopUser() {
   running = false;
}

@SuppressWarnings({"override", "UseSpecificCatch"})
public void run() {

   int count = 1;

   try {

      while(running && count <= 5) {

         synchronized(this) {

            while(paused) {

               wait();
            }
         }
```

```
            System.out.println(userName +
                  " : Sending Message " + count);

            count++;

            Thread.sleep(1000);
          }
        }

        catch(Exception e) {

          System.out.println(e);
        }

        System.out.println(userName + " has stopped.");
    }
}

public class ChatSystem {

    @SuppressWarnings("UseSpecificCatch")
    public static void main(String[] args) {

        try {

            ChatUser user1 = new ChatUser("User1");
            ChatUser user2 = new ChatUser("User2");

            // Set priorities
            user1.setPriority(Thread.MAX_PRIORITY);
            user2.setPriority(Thread.MIN_PRIORITY);

            // Start threads
            user1.start();
            user2.start();
```

```
    // Check thread status
    System.out.println("User1 Alive: " +
            user1.isAlive());

    System.out.println("User2 Alive: " +
            user2.isAlive());

    // Pause user1
    Thread.sleep(3000);

    System.out.println("Pausing User1...");
    user1.pauseUser();

    // Resume user1
    Thread.sleep(3000);

    System.out.println("Resuming User1...");
    user1.resumeUser();

    // Wait for threads
    user1.join();
    user2.join();

    // Stop threads
    user1.stopUser();
    user2.stopUser();

    System.out.println("Chat System Finished");
}

catch(Exception e) {

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

```
    }
}
```

PR 11

- **Step 1: Create Database**

Run this in MySQL first:

```
CREATE DATABASE company;

USE company;

CREATE TABLE employee (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    department VARCHAR(50),
    salary DOUBLE
);
```

- **Step 2: Simple Java Program**

```
import java.sql.*;
import java.util.Scanner;

public class EmployeeApp {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String url = "jdbc:mysql://localhost:3306/company";
        String user = "root";
        String password = "your_password";

        try {
            // Load Driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Connect Database
            Connection con = DriverManager.getConnection(url, user, password);

            int choice;

            do {
                System.out.println("\n1. Add Employee");
                System.out.println("2. View Employees");
                System.out.println("3. Update Salary");
                System.out.println("4. Delete Employee");
                System.out.println("5. Exit");

                System.out.print("Enter choice: ");
                choice = sc.nextInt();

                // ADD EMPLOYEE
                if (choice == 1) {

                    System.out.print("Enter Name: ");
                    String name = sc.next();

                    System.out.print("Enter Department: ");
                    String dept = sc.next();
```

```
    System.out.print("Enter Salary: ");
    double salary = sc.nextDouble();

    String query =
    "INSERT INTO employee(name, department, salary) VALUES(?,?,?)";

    PreparedStatement ps = con.prepareStatement(query);

    ps.setString(1, name);
    ps.setString(2, dept);
    ps.setDouble(3, salary);

    ps.executeUpdate();

    System.out.println("Employee Added!");

}

// VIEW EMPLOYEES
else if (choice == 2) {

    String query = "SELECT * FROM employee";

    Statement st = con.createStatement();

    ResultSet rs = st.executeQuery(query);

    while (rs.next()) {

        System.out.println(
                rs.getInt("id") + " " +
                rs.getString("name") + " " +
                rs.getString("department") + " " +
                rs.getDouble("salary")
        );
    }
}

// UPDATE EMPLOYEE
else if (choice == 3) {

    System.out.print("Enter Employee ID: ");
    int id = sc.nextInt();

    System.out.print("Enter New Salary: ");
    double salary = sc.nextDouble();

    String query =
    "UPDATE employee SET salary=? WHERE id=?";

    PreparedStatement ps = con.prepareStatement(query);

    ps.setDouble(1, salary);
    ps.setInt(2, id);

    ps.executeUpdate();

    System.out.println("Salary Updated!");
}

// DELETE EMPLOYEE
else if (choice == 4) {

    System.out.print("Enter Employee ID: ");
    int id = sc.nextInt();

    String query =
    "DELETE FROM employee WHERE id=?";
```

```
                PreparedStatement ps = con.prepareStatement(query);
                ps.setInt(1, id);
                ps.executeUpdate();
                System.out.println("Employee Deleted!");
            }
        } while (choice != 5);
        con.close();
      } catch (Exception e) {
        System.out.println(e);
      }
    }
}
```

PR 12.

```
import java.io.*;
import java.util.*;

public class PatientAnalysis {

   public static void main(String[] args) {

      ArrayList<Integer> hr = new ArrayList<>();

      try {

         BufferedReader br =
            new BufferedReader(new FileReader("patients.csv"));

         // Skip header
         String line = br.readLine();

         while((line = br.readLine()) != null) {

            String[] data = line.split(",");
```

```
    int heartRate = Integer.parseInt(data[2]);


    hr.add(heartRate);
}


br.close();


// Mean
int sum = 0;
```

```
for(int x : hr) sum += x;

double mean = (double) sum / hr.size();

// Median
Collections.sort(hr);

double median;

int n = hr.size();

if(n % 2 == 0) {
   median = (hr.get(n/2 - 1) + hr.get(n/2)) / 2.0;
} else {
   median = hr.get(n/2);
}

// Standard Deviation
double variance = 0;

for(int x : hr) {
   variance += Math.pow(x - mean, 2);
}

variance = variance / hr.size();

double stdDev = Math.sqrt(variance);

// Output
System.out.println("Mean = " + mean);
System.out.println("Median = " + median);
System.out.println("Std Dev = " + stdDev);

// (Optional) Bar Chart using JFreeChart
System.out.println("Bar chart can be generated using JFreeChart library.");
```

```
        }


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