Q-1. Write a Java program to create a class called "Rectangle" with width and height attributes. Calculate the area and perimeter, diagonal of the rectangle.

**POGRAM :**

// Q: 1

// AUTHOR: ARGHA DIGAR

// TITLE: Rectangle Class with Area, Perimeter, and Diagonal Calculation

// DESCRIPTION: This Java program defines a Rectangle class with width and height attributes and calculates the area, perimeter, and diagonal of the rectangle.

import java.lang.Math;

class Rectangle {

    private double width;

    private double height;

    public Rectangle(double wid, double heig) {

      width = wid;

      height = heig;

    }

    public double getWidth() {

        return width;

    }

    public void setWidth(double wid) {

    width = wid;

    }

    public double getHeight() {

        return height;

    }

    public void setHeight(double heig) {

    height = heig;

    }

    public double calculateArea() {

        return width \* height;

    }

    public double calculatePerimeter() {

        return 2 \* (width + height);

    }

    public double calculateDiagonal() {

        return Math.sqrt(width \* width + height \* height);

    }

}

public class Main {

    public static void main(String[] args) {

        // Create a Rectangle object with width 5.0 and height 7.0

        Rectangle rectangle = new Rectangle(5.0, 7.0);

        // Calculate and print the area, perimeter, and diagonal

        System.out.println("Rectangle Area: " + rectangle.calculateArea());

        System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());

        System.out.println("Rectangle Diagonal: " + rectangle.calculateDiagonal());

    }

}

**OUTPUT :**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\Rectangle>javac Main.java**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\Rectangle>java Main**

**Rectangle Area: 35.0**

**Rectangle Perimeter: 24.0**

**Rectangle Diagonal: 8.602325267042627**

Q-2. Write a Java program to create a class called "Circle" with a radius attribute. You can access and modify this attribute. Calculate the area and circumference of the circle.

**POGRAM :**

// Q: 2

// AUTHOR: ARGHA DIGAR

// TITLE: Circle Class with Area and Circumference Calculation

// DESCRIPTION: This Java program defines a Circle class with a radius attribute. It provides methods to access and modify the radius and calculates the area and circumference of the circle.

import java.lang.Math;

class Circle {

    private double radius;

    public Circle(double radiu) {

       radius = radiu;

    }

    public double getRadius() {

        return radius;

    }

    public void setRadius(double radiu) {

        radius = radiu;

    }

    public double calculateArea() {

        return Math.PI \* radius \* radius;

    }

    public double calculateCircumference() {

        return 2 \* Math.PI \* radius;

    }

}

public class Main {

    public static void main(String[] args) {

        // Create a Circle object with a radius of 3.0

        Circle circle = new Circle(3.0);

        // Access and modify the radius

        double newRadius = 5.0;

        circle.setRadius(newRadius);

        // Calculate and print the area and circumference

        System.out.println("Circle Radius: " + circle.getRadius());

        System.out.println("Circle Area: " + circle.calculateArea());

        System.out.println("Circle Circumference: " + circle.calculateCircumference());

    }

}

**OUTPUT :**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\Circle>javac Main.java**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\Circle>java Main**

**Circle Radius: 5.0**

**Circle Area: 78.53981633974483**

**Circle Circumference: 31.41592653589793**

Q-4. Write a Java program to create a class called "Airplane" with a flight number, destination, and departure time attributes, and methods to check flight status and delay.

**POGRAM :**

// Q: 4

// AUTHOR: ARGHA DIGAR

// TITLE: Airplane Class with Flight Status and Delay Methods

// DESCRIPTION: This Java program defines an Airplane class with flight number, destination, and departure time attributes. It provides methods to check flight status and delay the flight.

import java.text.SimpleDateFormat;

import java.util.Date;

class Airplane {

    private String flightNumber;

    private String destination;

    private Date departureTime;

    private boolean delayed;

    public Airplane(String flightNumber, String destination, Date departureTime) {

        this.flightNumber = flightNumber;

        this.destination = destination;

        this.departureTime = departureTime;

        this.delayed = false;

    }

    public String getFlightNumber() {

        return flightNumber;

    }

    public String getDestination() {

        return destination;

    }

    public Date getDepartureTime() {

        return departureTime;

    }

    public boolean isDelayed() {

        return delayed;

    }

    public void delayFlight() {

        delayed = true;

    }

    public void checkFlightStatus() {

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

        String status = delayed ? "Delayed" : "On time";

        System.out.println("Flight " + flightNumber + " to " + destination + " departing at " + sdf.format(departureTime) + " is " + status + ".");

    }

}

public class Main {

    public static void main(String[] args) {

        // Create an Airplane object for a flight

        Date departureTime = new Date(); // Current date and time

        Airplane flight = new Airplane("AA123", "New York", departureTime);

        // Check and print the initial flight status

        flight.checkFlightStatus();

        // Delay the flight and check status again

        flight.delayFlight();

        flight.checkFlightStatus();

    }

}

**OUTPUT :**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\Airplane>javac Main.java**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\Airplane>java Main**

**Flight AA123 to New York departing at 2023-09-15 21:59:35 is On time.**

**Flight AA123 to New York departing at 2023-09-15 21:59:35 is Delayed.**

Q-5. Write a Java program to create a class called "School" with attributes for students, teachers, and classes, and methods to add and remove students and teachers, and to create classes.

**POGRAM :**

// Q: 5

// AUTHOR: ARGHA DIGAR

// TITLE: Java School Management System

// DESCRIPTION: This Java program demonstrates a simple school management system that allows you to add and remove students and teachers, as well as create classes in a school.

class School {

     Student[] students;

     Teacher[] teachers;

     Classroom[] classrooms;

     int studentCount;

     int teacherCount;

     int classroomCount;

     School(int maxStudents, int maxTeachers, int maxClassrooms) {

        students = new Student[maxStudents];

        teachers = new Teacher[maxTeachers];

        classrooms = new Classroom[maxClassrooms];

        studentCount = 0;

        teacherCount = 0;

        classroomCount = 0;

    }

    // Add a student to the school

    public void addStudent(Student student) {

        if (studentCount < students.length) {

            students[studentCount] = student;

            studentCount++;

        } else {

            System.out.println("Maximum student limit reached.");

        }

    }

    // Remove a student from the school

    public void removeStudent(Student student) {

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

            if (students[i] == student) {

                for (int j = i; j < studentCount - 1; j++) {

                    students[j] = students[j + 1];

                }

                students[studentCount - 1] = null;

                studentCount--;

                return;

            }

        }

        System.out.println("Student not found.");

    }

    // Add a teacher to the school

    public void addTeacher(Teacher teacher) {

        if (teacherCount < teachers.length) {

            teachers[teacherCount] = teacher;

            teacherCount++;

        } else {

            System.out.println("Maximum teacher limit reached.");

        }

    }

    // Remove a teacher from the school

    public void removeTeacher(Teacher teacher) {

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

            if (teachers[i] == teacher) {

                for (int j = i; j < teacherCount - 1; j++) {

                    teachers[j] = teachers[j + 1];

                }

                teachers[teacherCount - 1] = null;

                teacherCount--;

                return;

            }

        }

        System.out.println("Teacher not found.");

    }

    // Create a new class and add it to the school

    public void createClass(String className) {

        if (classroomCount < classrooms.length) {

            classrooms[classroomCount] = new Classroom(className);

            classroomCount++;

        } else {

            System.out.println("Maximum classroom limit reached.");

        }

    }

  // Display information about the school

    public void displaySchoolInfo() {

    System.out.println(":::::::::::::::School Information:::::::::::::::");

    System.out.println("Number of Students: " + studentCount);

    System.out.println("Students:");

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

        System.out.println("  " + (i + 1) + ". " + students[i].getName());

    }

    System.out.println("Number of Teachers: " + teacherCount);

    System.out.println("Teachers:");

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

        System.out.println("  " + (i + 1) + ". " + teachers[i].getName());

    }

    System.out.println("Number of Classes: " + classroomCount);

    System.out.println("Classes:");

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

        System.out.println("  " + (i + 1) + ". " + classrooms[i].getClassName());

    }

}

    // Inner class to represent a Student

    public class Student {

        String name;

        public Student(String nam) {

            name = nam;

        }

        public String getName() {

            return name;

        }

    }

    // Inner class to represent a Teacher

    public class Teacher {

        private String name;

        public Teacher(String nam) {

            name = nam;

        }

        public String getName() {

            return name;

        } }

    // Inner class to represent a Classroom

    public class Classroom {

        private String className;

        public Classroom(String classNam) {

           className = classNam;

        }

        public String getClassName() {

            return className;

        }

    }

}

public class Main {

    public static void main(String[] args) {

        School school = new School(100, 20, 10);

        // Adding students and teachers

        school.addStudent(school.new Student("Argha"));

        school.addStudent(school.new Student("Soukat"));

        school.addStudent(school.new Student("Sudipa"));

        school.addStudent(school.new Student("Sanjana"));

        school.addStudent(school.new Student("Kuntal"));

        school.addTeacher(school.new Teacher("Mr. Avijit"));

        school.addTeacher(school.new Teacher("Ms. rumpa"));

         school.addTeacher(school.new Teacher("Mr. Preetam"));

        // Creating classes

        school.createClass("java");

        school.createClass("DSA");

        // Display school information

        school.displaySchoolInfo();

    }

}

**OUTPUT :**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\School>javac Main.java**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\School>java Main**

**:::::::::::::::School Information:::::::::::::::**

**Number of Students: 5**

**Students:**

**1. Argha**

**2. Soukat**

**3. Sudipa**

**4. Sanjana**

**5. Kuntal**

**Number of Teachers: 3**

**Teachers:**

**1. Mr. Avijit**

**2. Ms. rumpa**

**3. Mr. Preetam**

**Number of Classes: 2**

**Classes:**

**1. java**

**2. DSA**

Q-6. Write a Java program to create a class called "Movie" with attributes for title, director, actors, and reviews, and methods for adding and retrieving reviews.

**POGRAM :**

// Q: 5

// AUTHOR: ARGHA DIGAR

// TITLE: Movie Class

// DESCRIPTION: This Java program defines a simple Movie class with attributes for a title, director, actor, and review, along with methods for adding and retrieving reviews.

class Movie {

     String title;

     String director;

     String actor;

     String review;

    public Movie(String title, String director, String actor, String reviewText) {

        this.title = title;

        this.director = director;

        this.actor = actor;

        this.review = reviewText;

    }

}

public class Main {

    public static void main(String[] args) {

        Movie movie = new Movie("Harry Potter", "Mike Newell", "Daniel Radcliffe","Great performances by the actors.");

        System.out.println("Movie Title: " + movie.title);

        System.out.println("Director: " + movie.director);

        System.out.println("Actor: " + movie.actor);

        System.out.println("Review: " + movie.review);

    }

}

**OUTPUT :**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\MOVIE>javac Main.java**

**C:\Users\Argha Digar\Desktop\JAVA\E2L\ASSIGEMENT\ASS2\MOVIE>java Main**

**Movie Title: Harry Potter**

**Director: Mike Newell**

**Actor: Daniel Radcliffe**

**Review: Great performances by the actors.**