DAY-10

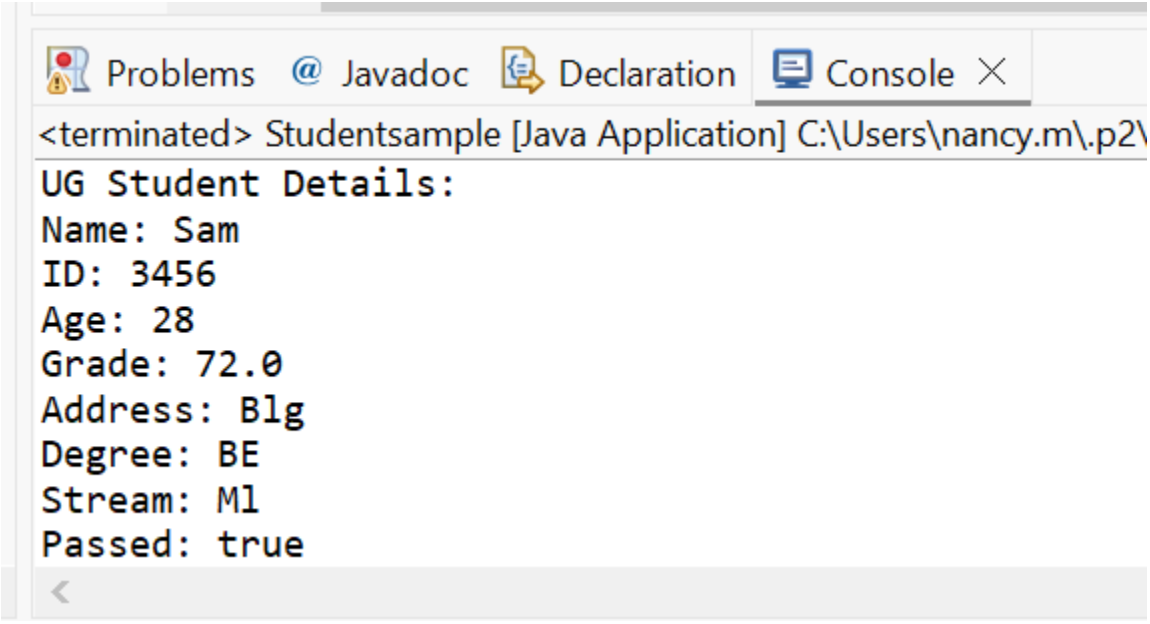
Date:22.07.2025
Name:Nancy M

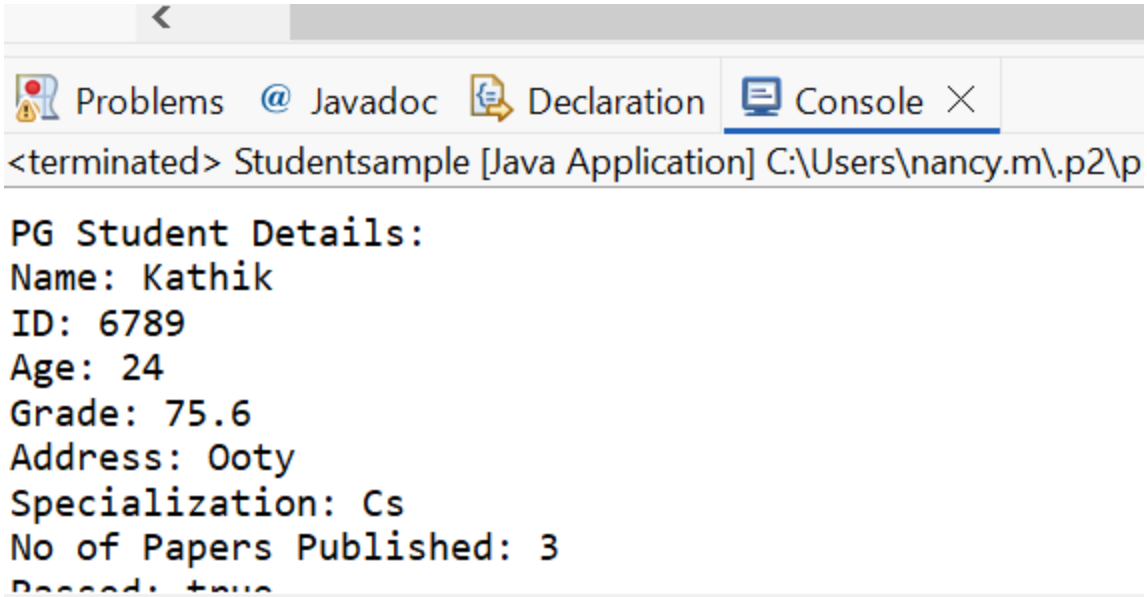
1)Create a class named Student.Include the following protected member variables.
name,id,age,grade,address
Include appropriate getters and setters
Include a default constructor and a 5-argument constructor. The order of arguments in the 5- argument constructor is name, id, age, grade and address.
Include the following public methods in the Student class.
void display()
Display the details of the student.
boolean isPassed()
A student is said to have passed if his/her grade is above 50. This method returns true if the student has passed.
Create a subclass of Student named UGStudent .
Include the following private member variables.
Degree stream
Include appropriate getters / setters
Include a default constructor and a 7-argument constructor. The order of arguments in the 7-argument constructor is name, id, age, grade, address, degree and stream.
Include the following public methods in the UGStudent class.
void display()
Display the details of the ug student in the format as specified in the output.
boolean isPassed()
A ug student is said to have passed if his/her grade is above 70. This method returns true if the student has passed.
Create another subclass of Student named PGStudent .
Include the following private member variables.
specialization
noOfPapersPublished
Include appropriate getters / setters
Include a default constructor and a 7-argument constructor. The order of arguments in the 7-argument constructor is name, id, age, grade, address, specialization and number of papers published.
Include the following public methods in the PGStudent class.
void display()
Display the details of the pg student in the format as specified in the output.
boolean isPassed()
A pg student is said to have passed if his/her grade is above 70 and if he/she has published atleast 2 papers.This method returns true if the student has passed.
Create a class called Main to test the above classes.

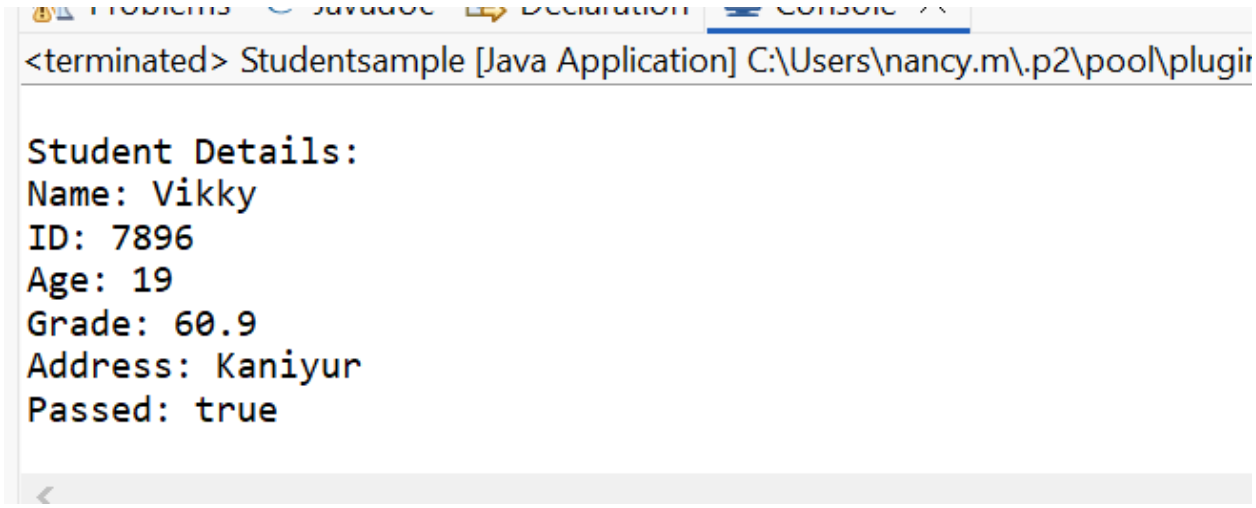
```
package com.training.ooc;
public class Studentsample {

   static class Student {
      protected String name;
      protected String id;
      protected int age;
      protected double grade;
      protected String address;

      public Student() {}

      public Student(String name, String id, int age, double grade, String address) {
         this.name = name;
         this.id = id;
         this.age = age;
         this.grade = grade;
         this.address = address;
      }

      public String getName() { return name; }
      public void setName(String name) { this.name = name; }
      public String getId() { return id; }
      public void setId(String id) { this.id = id; }
      public int getAge() { return age; }
      public void setAge(int age) { this.age = age; }
      public double getGrade() { return grade; }
      public void setGrade(double grade) { this.grade = grade; }
      public String getAddress() { return address; }
      public void setAddress(String address) { this.address = address; }

      public void display() {
         System.out.println("Student Details:");
         System.out.println("Name: " + name);
         System.out.println("ID: " + id);
         System.out.println("Age: " + age);
         System.out.println("Grade: " + grade);
         System.out.println("Address: " + address);
      }

      public boolean isPassed() {
         return grade > 50;
      }
   }

   static class UGStudent extends Student {
      private String degree;
      private String stream;
```

```
    public UGStudent() {}
    public UGStudent(String name, String id, int age, double grade, String address, String degree, String
stream) {
        super(name, id, age, grade, address);
        this.degree = degree;
        this.stream = stream;
    }
    public String getDegree() { return degree; }
    public void setDegree(String degree) { this.degree = degree; }
    public String getStream() { return stream; }
    public void setStream(String stream) { this.stream = stream; }

    public void display() {
        System.out.println("UG Student Details:");
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("Age: " + age);
        System.out.println("Grade: " + grade);
        System.out.println("Address: " + address);
        System.out.println("Degree: " + degree);
        System.out.println("Stream: " + stream);
    }

    public boolean isPassed() {
        return grade > 70;
    }
  }

  static class PGStudent extends Student {
    private String specialization;
    private int noOfPapersPublished;
    public PGStudent() {}
    public PGStudent(String name, String id, int age, double grade, String address, String specialization,
int noOfPapersPublished) {
        super(name, id, age, grade, address);
        this.specialization = specialization;
        this.noOfPapersPublished = noOfPapersPublished;
    }
    public String getSpecialization() { return specialization; }
    public void setSpecialization(String specialization) { this.specialization = specialization; }
    public int getNoOfPapersPublished() { return noOfPapersPublished; }
    public void setNoOfPapersPublished(int count) { this.noOfPapersPublished = count; }
    public void display() {
        System.out.println("PG Student Details:");
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("Age: " + age);
        System.out.println("Grade: " + grade);
        System.out.println("Address: " + address);
```

```
        System.out.println("Specialization: " + specialization);
        System.out.println("No of Papers Published: " + noOfPapersPublished);
    }

    public boolean isPassed() {
        return grade > 70 && noOfPapersPublished >= 2;
    }
  }
   public static void main(String[] args) {
    UGStudent ug = new UGStudent("Sam", "3456", 28, 72, "Blg", "BE", "Ml");
    ug.display();
    System.out.println("Passed: " + ug.isPassed());
    System.out.println();
    PGStudent pg = new PGStudent("Kathik", "6789", 24, 75.6, "Ooty", "Cs", 3);
    pg.display();
    System.out.println("Passed: " + pg.isPassed());
    System.out.println();
    Student s = new Student("Vikky", "7896", 19, 60.9, "Kaniyur");
    s.display();
    System.out.println("Passed: " + s.isPassed());
  }
}
```







2) The task is to get the details of the vehicle and display the details using a menu driven application.
Write a Java program to Implement this task.
Create a class Vehicle
Include the following protected data members / attributes:
make – of type String
vehicleNumber – of type String
fuelType – of type String
fuelCapacity - of type Integer
cc – of type Integer

Include the following public methods
Create a constructor that initializes all the data members --- public Vehicle(String make,String vehicleNumber,String fuelType,Integer fuelCapacity,Integer cc)
displayMake – Display the make of the vehicle
"displayBasicInfo" – display basic information of the vehicle.
"displayDetailInfo" – An empty method.

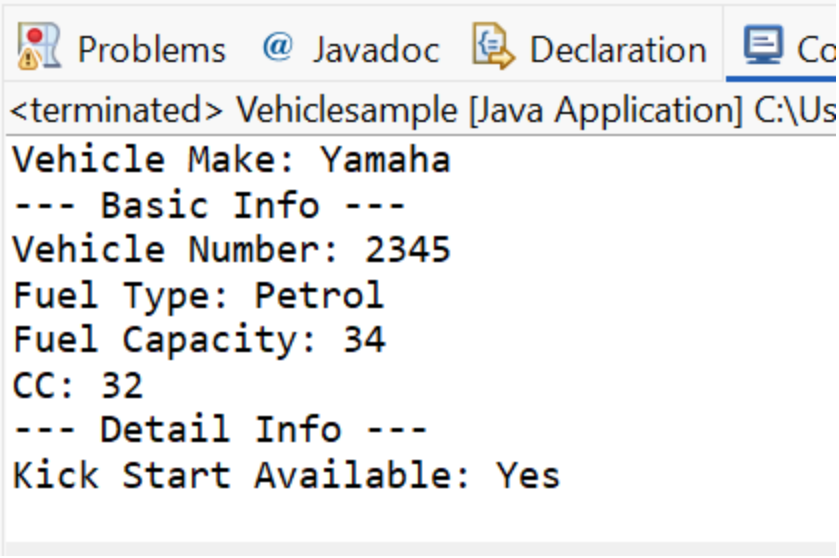
Create a class TwoWheeler that extends Vehicle
kickStartAvailable – of type Boolean.
"displayDetailInfo" – displays the availability of kick start.
Create a class FourWheeler that extends Vehicle
audioSystem – of type String.
numberOfDoors – of type Integer.

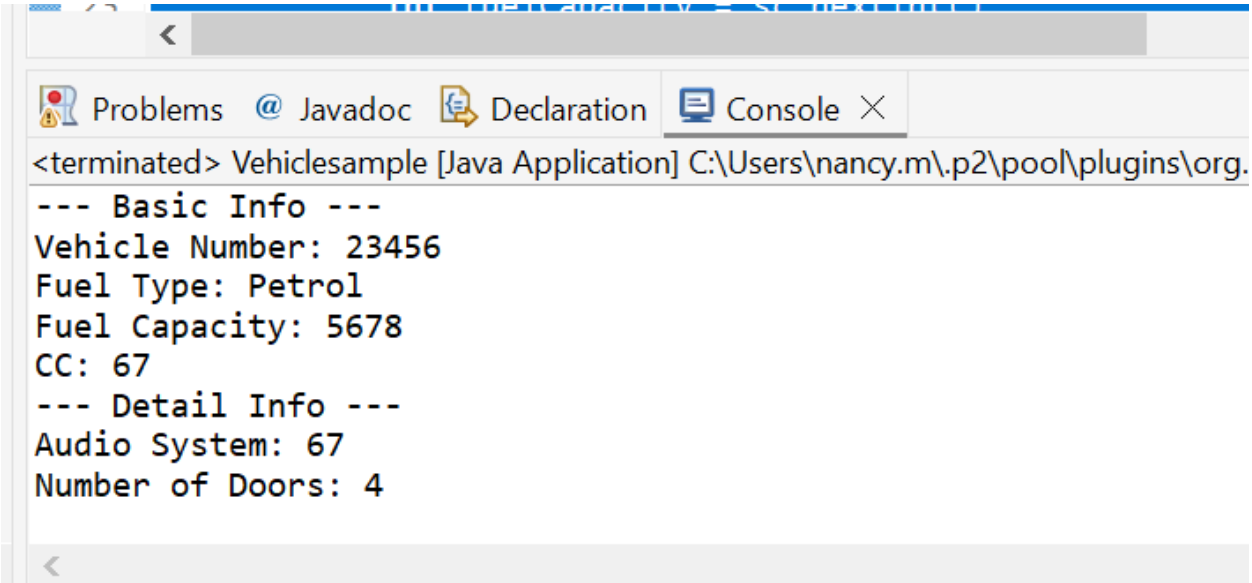
"displayDetailInfo" - displays the audio system and number of doors.
displayDetailInfo - overridden method
Include getter setters for all the classes.
Create a main class to test the classes defined above.

```
package com.training.ooc;
import java.util.Scanner;
public class Vehiclesample {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("1. 2 Wheeler");
		System.out.println("2. 4 Wheeler");
		System.out.print("Enter Vehicle Type: ");
		int type = sc.nextInt();
		sc.nextLine();
		System.out.print("Vehicle Make: ");
		String make = sc.nextLine();
		System.out.print("Vehicle Number: ");
		String vehicleNumber = sc.nextLine();
		System.out.print("Fuel Type (Petrol/Diesel): ");
		String fuelType = sc.nextLine();
		System.out.print("Fuel Capacity: ");
		int fuelCapacity = sc.nextInt();
		System.out.print("Engine CC: ");
		int cc = sc.nextInt();
		sc.nextLine();
		if (type == 1) {
		   System.out.print("Kick Start Available (true/false): ");
		   boolean kickStartAvailable = sc.nextBoolean();
		   TwoWheeler bike = new TwoWheeler(make, vehicleNumber, fuelType, fuelCapacity, cc,
kickStartAvailable);
		   System.out.println("\n--- Vehicle Make ---");
```

```
                bike.displayMake();
                System.out.println("--- Basic Info ---");
                bike.displayBasicInfo();
                System.out.println("--- Detail Info ---");
                bike.displayDetailInfo();
            } else if (type == 2) {
                System.out.print("Audio System: ");
                String audioSystem = sc.nextLine();
                System.out.print("Number of Doors: ");
                int numberOfDoors = sc.nextInt();
                FourWheeler car = new FourWheeler(make, vehicleNumber, fuelType, fuelCapacity, cc,
audioSystem, numberOfDoors);
                System.out.println("\n--- Vehicle Make ---");
                car.displayMake();
                System.out.println("--- Basic Info ---");
                car.displayBasicInfo();
                System.out.println("--- Detail Info ---");
                car.displayDetailInfo();
            } else {
                System.out.println("Invalid Vehicle Type.");
            }
            sc.close();
        }
    }

    class Vehicle {
        protected String make;
        protected String vehicleNumber;
        protected String fuelType;
        protected int fuelCapacity;
        protected int cc;
        public Vehicle(String make, String vehicleNumber, String fuelType, int fuelCapacity, int cc) {
            this.make = make;
            this.vehicleNumber = vehicleNumber;
            this.fuelType = fuelType;
            this.fuelCapacity = fuelCapacity;
            this.cc = cc;
        }
        public String getMake() { return make; }
        public void setMake(String make) { this.make = make; }
        public String getVehicleNumber() { return vehicleNumber; }
        public void setVehicleNumber(String vehicleNumber) { this.vehicleNumber = vehicleNumber; }
        public String getFuelType() { return fuelType; }
        public void setFuelType(String fuelType) { this.fuelType = fuelType; }
        public int getFuelCapacity() { return fuelCapacity; }
        public void setFuelCapacity(int fuelCapacity) { this.fuelCapacity = fuelCapacity; }
        public int getCc() { return cc; }
        public void setCc(int cc) { this.cc = cc; }
        public void displayMake() {
```

```
            System.out.println("Vehicle Make: " + make);
        }
        public void displayBasicInfo() {
            System.out.println("Vehicle Number: " + vehicleNumber);
            System.out.println("Fuel Type: " + fuelType);
            System.out.println("Fuel Capacity: " + fuelCapacity);
            System.out.println("CC: " + cc);
        }
        public void displayDetailInfo() {

        }
    }

    class TwoWheeler extends Vehicle {
        private boolean kickStartAvailable;
        public TwoWheeler(String make, String vehicleNumber, String fuelType, int fuelCapacity, int cc,
boolean kickStartAvailable) {
            super(make, vehicleNumber, fuelType, fuelCapacity, cc);
            this.kickStartAvailable = kickStartAvailable;
        }
        public boolean isKickStartAvailable() { return kickStartAvailable; }
        public void setKickStartAvailable(boolean kickStartAvailable) { this.kickStartAvailable =
kickStartAvailable; }
        @Override
        public void displayDetailInfo() {
            System.out.println("Kick Start Available: " + (kickStartAvailable ? "Yes" : "No"));
        }
    }
    class FourWheeler extends Vehicle {
        private String audioSystem;
        private int numberOfDoors;
        public FourWheeler(String make, String vehicleNumber, String fuelType, int fuelCapacity, int
cc, String audioSystem, int numberOfDoors) {
            super(make, vehicleNumber, fuelType, fuelCapacity, cc);
            this.audioSystem = audioSystem;
            this.numberOfDoors = numberOfDoors;
        }
        public String getAudioSystem() { return audioSystem; }
        public void setAudioSystem(String audioSystem) { this.audioSystem = audioSystem; }
        public int getNumberOfDoors() { return numberOfDoors; }
        public void setNumberOfDoors(int numberOfDoors) { this.numberOfDoors = numberOfDoors; }

        public void displayDetailInfo() {
            System.out.println("Audio System: " + audioSystem);
            System.out.println("Number of Doors: " + numberOfDoors);
        }
    }
```





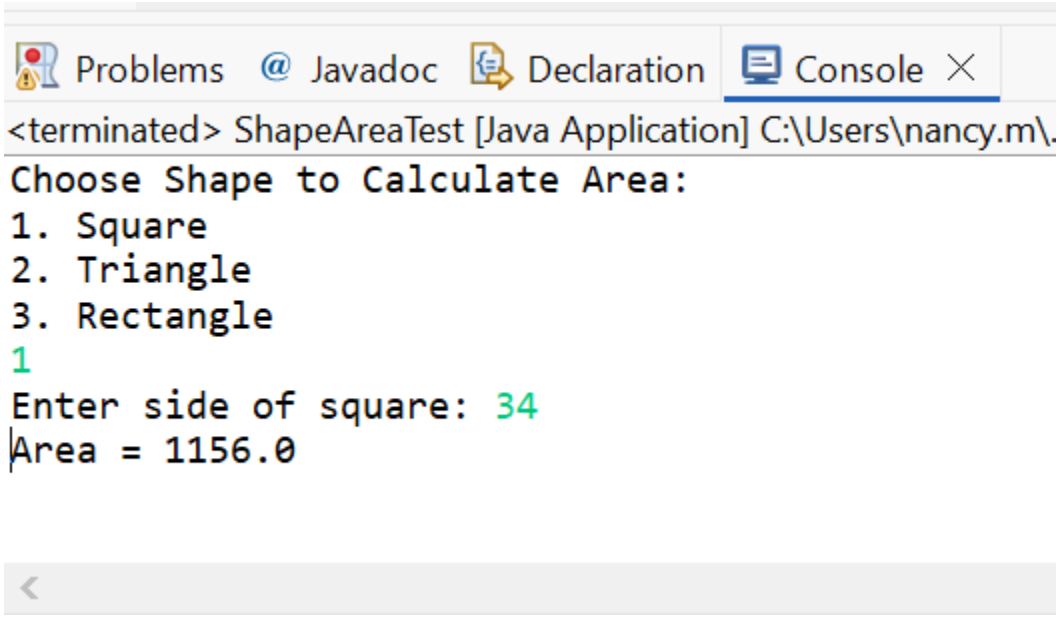
3.Create a class Shape and inherit three classes Square,Triangle and Rectangle. Implement the
method double calculateArea() in Shape class and override the method in the subclasses. Use runtime polymorphism to call the calculateArea() method.

```
package com.training.ooc;
import java.util.Scanner;
abstract class Shape {
   abstract double calculateArea();
}
class Square extends Shape {
   private double side;
```

```
    public Square(double side) {
        this.side = side;
    }
    public double calculateArea() {
        return side * side;
    }
    public double getSide() {
        return side;
    }
    public void setSide(double side) {
        this.side = side;
    }
}
class Triangle extends Shape {
    private double base;
    private double height;
    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }
    public double calculateArea() {
        return 0.5 * base * height;
    }
    public double getBase() {
        return base;
    }
    public double getHeight() {
        return height;
    }
    public void setBase(double base) {
        this.base = base;
    }
    public void setHeight(double height) {
        this.height = height;
    }
}
// Rectangle class
class Rectangle extends Shape {
    private double length;
    private double breadth;
    public Rectangle(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }
    public double calculateArea() {
        return length * breadth;
    }
    public double getLength() {
        return length;
```

```
    }
    public double getBreadth() {
        return breadth;
    }
    public void setLength(double length) {
        this.length = length;
    }
    public void setBreadth(double breadth) {
        this.breadth = breadth;
    }
}
// Main class
public class ShapeAreaTest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Shape shape;
        System.out.println("Choose Shape to Calculate Area:");
        System.out.println("1. Square\n2. Triangle\n3. Rectangle");
        int choice = sc.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter side of square: ");
                double side = sc.nextDouble();
                shape = new Square(side);
                break;
            case 2:
                System.out.print("Enter base of triangle: ");
                double base = sc.nextDouble();
                System.out.print("Enter height of triangle: ");
                double height = sc.nextDouble();
                shape = new Triangle(base, height);
                break;
            case 3:
                System.out.print("Enter length of rectangle: ");
                double length = sc.nextDouble();
                System.out.print("Enter breadth of rectangle: ");
                double breadth = sc.nextDouble();
                shape = new Rectangle(length, breadth);
                break;
            default:
                System.out.println("Invalid choice");
                sc.close();
                return;
        }
        // Runtime Polymorphism
        System.out.println("Area = " + shape.calculateArea());
        sc.close();
    }
```

}



4.A Company provides an initial training for all its employees, once they join the company. During the training phase they call the employees as "Associate". The initial training is conducted for 60 days for each Associate. In these 60 days they learn various technologies. The first 20 days they learn "Core skills", the next 20 days they learn "Advanced modules" and the final 20 days they go to the "Project phase". Help the Company to find in which phase the associates are in.
Create a class Associate with associateId(int),associateName(String),workStatus(String). Include getters and setters and constructors.
Add a method trackAssociateStatus
- This method takes the number of days as argument and sets the work status of the associate based on the number of days. If the number of days is greater than 60 days then set the work status as "Deployed in project".
In the Main class, create an object for the Associate class; Get the details assign the value for its attributes using the setters. Invoke the trackAssociateStatus method and find the work status and display the details.

```
package com.training.ooc;
import java.util.Scanner;
class Associate {
   private int associateId;
   private String associateName;
   private String workStatus;

   public Associate(int associateId, String associateName) {
      this.associateId = associateId;
```

```
        this.associateName = associateName;
        this.workStatus = "Not Started";
    }

    public int getAssociateId() {
        return associateId;
    }
    public void setAssociateId(int associateId) {
        this.associateId = associateId;
    }
    public String getAssociateName() {
        return associateName;
    }
    public void setAssociateName(String associateName) {
        this.associateName = associateName;
    }
    public String getWorkStatus() {
        return workStatus;
    }
    public void setWorkStatus(String workStatus) {
        this.workStatus = workStatus;
    }
    public void trackAssociateStatus(int noOfDays) {
        if (noOfDays <= 20) {
            setWorkStatus("Core skills");
        } else if (noOfDays <= 40) {
            setWorkStatus("Advanced modules");
        } else if (noOfDays <= 60) {
            setWorkStatus("Project phase");
        } else {
            setWorkStatus("Deployed in project");
        }
    }
    public void displayDetails() {
        System.out.println("Associate ID: " + associateId);
        System.out.println("Associate Name: " + associateName);
        System.out.println("Work Status: " + workStatus);
    }
}
public class Associatesample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Associate ID: ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Associate Name: ");
        String name = sc.nextLine();
        System.out.print("Enter number of days: ");
```

```
        int days = sc.nextInt();

        Associate associate = new Associate(id, name);

        associate.trackAssociateStatus(days);

        System.out.println("\nAssociate Details:");
        associate.displayDetails();
        sc.close();
    }
}
```