NAME: Pavithra R
DATE: 22/07/2025

DAY 12 ASSESSMENT

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.

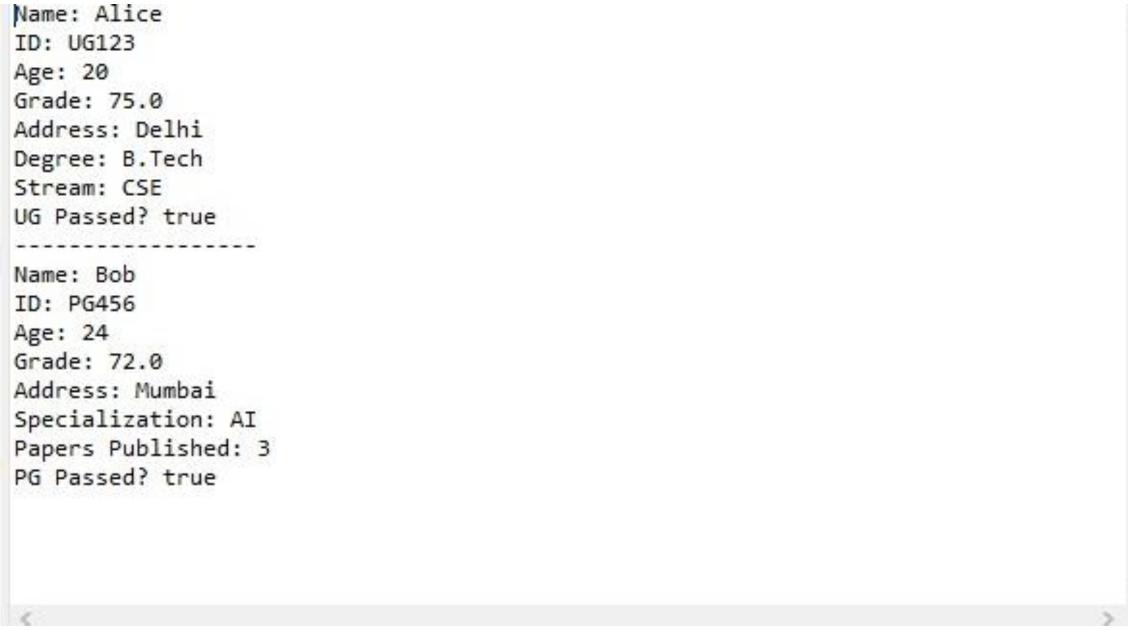
CODE:

```
package Student;
```

```
public 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("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;
  }


        public static void main(String args[]) {
                UGStudent ug = new UGStudent("Alice", "UG123", 20, 75, "Delhi", "B.Tech", "CSE");
    ug.display();
    System.out.println("UG Passed? " + ug.isPassed());
    System.out.println("------------------");
    PGStudent pg = new PGStudent("Bob", "PG456", 24, 72, "Mumbai", "AI", 3);
    pg.display();
    System.out.println("PG Passed? " + pg.isPassed());
        }
}
class UGStudent extends Student {
  private String degree;
```

```java
    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() {
        super.display();
        System.out.println("Degree: " + degree);
        System.out.println("Stream: " + stream);
    }
    public boolean isPassed() {
        return grade > 70;
    }
}
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 noOfPapersPublished) { this.noOfPapersPublished =
noOfPapersPublished; }
    public void display() {
        super.display();
        System.out.println("Specialization: " + specialization);
        System.out.println("Papers Published: " + noOfPapersPublished);
    }
    public boolean isPassed() {
        return grade > 70 && noOfPapersPublished >= 2;
    }
}
```

OUTPUT:



```
Name: Alice
ID: UG123
Age: 20
Grade: 75.0
Address: Delhi
Degree: B.Tech
Stream: CSE
UG Passed? true
------------------
Name: Bob
ID: PG456
Age: 24
Grade: 72.0
Address: Mumbai
Specialization: AI
Papers Published: 3
PG Passed? true
```

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.

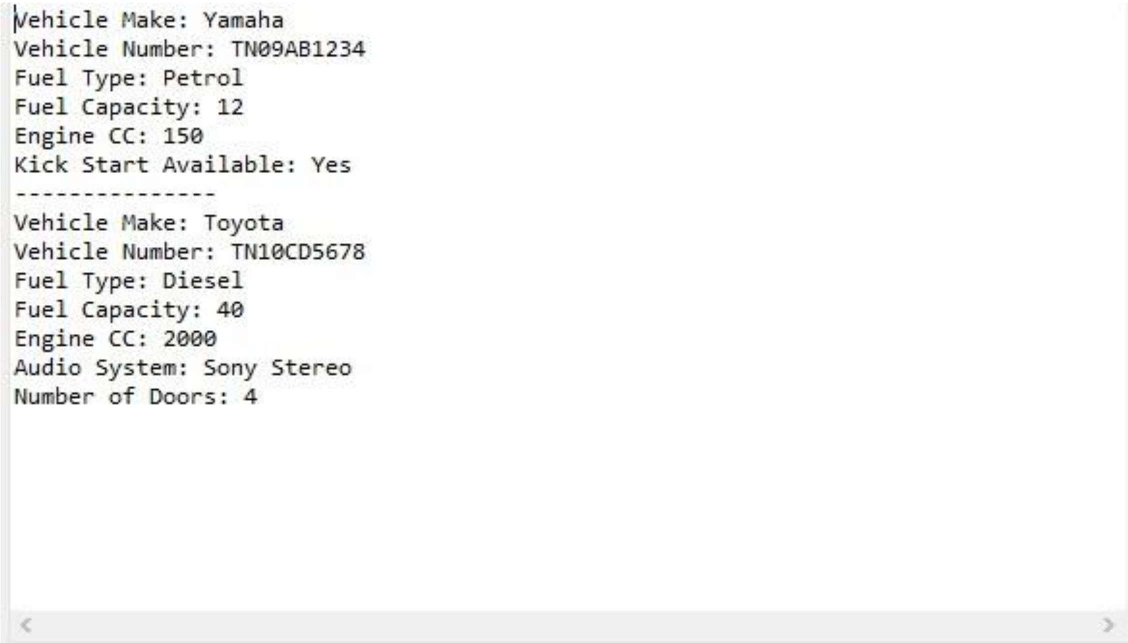
"displayDetailInfo" – displays the availability of kick start.
Create a class FourWheeler that extends Vehicle
audioSystem – of type String.
numberOfDoors – of type Integer.

"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.

CODE:

```
package Student;
public 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 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("Engine CC: " + cc);
  }
  public void displayDetailInfo() {

  }


  public static void main(String[] args) {
    Vehicle bike = new TwoWheeler("Yamaha", "TN09AB1234", "Petrol", 12, 150, true);
    bike.displayMake();
    bike.displayBasicInfo();
    bike.displayDetailInfo();
    System.out.println("---------------");
    Vehicle car = new FourWheeler("Toyota", "TN10CD5678", "Diesel", 40, 2000, "Sony Stereo", 4);
    car.displayMake();
    car.displayBasicInfo();
    car.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 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 void displayDetailInfo() {
    System.out.println("Audio System: " + audioSystem);
    System.out.println("Number of Doors: " + numberOfDoors);
  }
}
```

OUTPUT:



```
Vehicle Make: Yamaha
Vehicle Number: TN09AB1234
Fuel Type: Petrol
Fuel Capacity: 12
Engine CC: 150
Kick Start Available: Yes
---------------
Vehicle Make: Toyota
Vehicle Number: TN10CD5678
Fuel Type: Diesel
Fuel Capacity: 40
Engine CC: 2000
Audio System: Sony Stereo
Number of Doors: 4
```

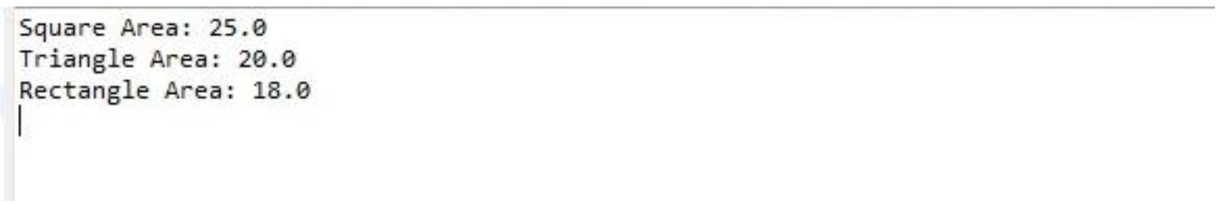
3) Create a class Shape and inherit three classes Square,Triangle and Rectangle. Implement the method double calculateArea() in Shape class and override the method in the subclasses. Use runtime polymorphism to call the calculateArea() method.

CODE:

```
package Student;
class Shape {
  public double calculateArea() {
     return 0.0;
  }

  public static void main(String[] args) {
     Shape s1 = new Square(5);
     Shape s2 = new Triangle(10, 4);
     Shape s3 = new Rectangle(6, 3);
     System.out.println("Square Area: " + s1.calculateArea());
     System.out.println("Triangle Area: " + s2.calculateArea());
     System.out.println("Rectangle Area: " + s3.calculateArea());
  }
}
```

```
class Square extends Shape {
  double side;
  public Square(double side) {
     this.side = side;
  }

  public double calculateArea() {
     return side * side;
  }
}
class Triangle extends Shape {
  double base, height;
  public Triangle(double base, double height) {
     this.base = base;
     this.height = height;
  }

  public double calculateArea() {
     return 0.5 * base * height;
  }
}
class Rectangle extends Shape {
  double length, width;
  public Rectangle(double length, double width) {
     this.length = length;
     this.width = width;
  }

  public double calculateArea() {
     return length * width;
  }
}
```

OUTPUT:



```
Square Area: 25.0
Triangle Area: 20.0
Rectangle Area: 18.0
```

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”.

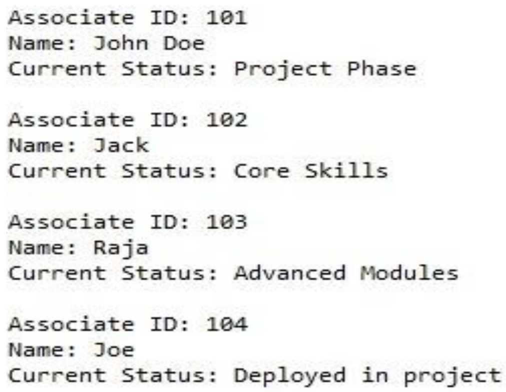
In the Main class, create an object for the Associate class; Get the details assign the value for its attributes using the setters. Invoke the trackAssociateStatus method and find the work status and display the details.

CODE:

```
package Student;
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 = "";
  }
  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 days) {
    if (days <= 20) {
      workStatus = "Core Skills";
    } else if (days <= 40) {
      workStatus = "Advanced Modules";
    } else if (days <= 60) {
      workStatus = "Project Phase";
    } else {
      workStatus = "Deployed in project";
    }
  }
  public void display() {
    System.out.println("Associate ID: " + associateId);
    System.out.println("Name: " + associateName);
```

```
        System.out.println("Current Status: " + workStatus);
    }

    public static void main(String[] args) {
        Associate a = new Associate(101, "John Doe");
        a.trackAssociateStatus(45);
        a.display();

        System.out.pri

        ntln();

        Associate a1 = new Associate(102, "Jack");
        a1.trackAssociateStatus(18);
        a1.display();

        System.out.pri

        ntln();

        Associate a2 = new Associate(103, "Raja");
        a2.trackAssociateStatus(25);
        a2.display();

        System.out.pri

        ntln();

        Associate a3 = new Associate(104, "Joe");
        a3.trackAssociateStatus(65);
        a3.display();
    }
}
```

Output:



```
Associate ID: 101
Name: John Doe
Current Status: Project Phase

Associate ID: 102
Name: Jack
Current Status: Core Skills

Associate ID: 103
Name: Raja
Current Status: Advanced Modules

Associate ID: 104
Name: Joe
Current Status: Deployed in project
```