Karnish N
22/07/2025

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.

CODING :

Student.java

```java
public class Student {
    protected String name;
    protected String id;
    protected int age;
    protected double grade;
    protected String address;


    public Student() {
        name = "";
        id = "";
        age = 0;
        grade = 0.0;
        address = "";
    }


    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; }
```

```java
    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;
    }
}
```

UGStudent.java

```java
public class UGStudent extends Student {
    private String degree;
    private String stream;


    public UGStudent() {
        super();
        degree = "";
        stream = "";
    }



    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; }
```

```java
    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;
    }
}
```

PGStudent.java

```java
public class PGStudent extends Student {
    private String specialization;
    private int noOfPapersPublished;


    public PGStudent() {
        super();
        specialization = "";
        noOfPapersPublished = 0;
    }


    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; }
```

```java
    public int getNoOfPapersPublished() { return noOfPapersPublished; }
    public void setNoOfPapersPublished(int noOfPapersPublished) {
this.noOfPapersPublished = noOfPapersPublished; }


    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;
    }
}
```

Main.java

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

        Student s = new Student("John", "S101", 20, 65.0, "Delhi");
        s.display();
        System.out.println("Passed: " + s.isPassed());
        System.out.println();

        UGStudent ug = new UGStudent("Ravi", "UG202", 19, 72.5,
"Chennai", "B.Tech", "IT");
        ug.display();
        System.out.println("Passed: " + ug.isPassed());
        System.out.println();
        PGStudent pg = new PGStudent("Meera", "PG303", 24, 78.0,
"Bangalore", "Data Science", 3);
```

```
        pg.display();
        System.out.println("Passed: " + pg.isPassed());

    }

}
```

OUTPUT:

```
PS C:\Users\karnish.n\Desktop\New folder> cd "c:\Users\karnish.n\Desktop\New folder\" ; if ($?) { javac Main.jav
a } ; if ($?) { java Main }
Name: John
ID: S101
Age: 20
Grade: 65.0
Address: Delhi
Passed: true

UG Student Details:
Name: Ravi
ID: UG202
Age: 19
Grade: 72.5
Address: Chennai
Degree: B.Tech
Stream: IT
Passed: true

PG Student Details:
Name: Meera
ID: PG303
Age: 24
Grade: 78.0
Address: Bangalore
Specialization: Data Science
No. of Papers Published: 3
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.

CODING :

```java
import java.util.*;

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; }
```

```java
    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("*** " + 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; }

    public void displayDetailInfo() {
```

```java
            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);
    }
}


public class VehicleMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("1. Two Wheeler");
        System.out.println("2. Four Wheeler");
        System.out.print("Enter Vehicle Type: ");
        int choice = sc.nextInt();
        sc.nextLine();
```

```java
        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("CC: ");
        int cc = sc.nextInt();
        sc.nextLine();

        Vehicle vehicle;

        if (choice == 1) {
            System.out.print("Kick Start Available (yes/no): ");
            String kick = sc.nextLine();
            boolean kickStartAvailable = kick.equalsIgnoreCase("yes");
            vehicle = new TwoWheeler(make, vehicleNumber, fuelType,
fuelCapacity, cc, kickStartAvailable);
        } else {
            System.out.print("Audio System: ");
            String audioSystem = sc.nextLine();

            System.out.print("Number of Doors: ");
            int numberOfDoors = sc.nextInt();
            vehicle = new FourWheeler(make, vehicleNumber, fuelType,
fuelCapacity, cc, audioSystem, numberOfDoors);
        }

        vehicle.displayMake();
        vehicle.displayBasicInfo();
        vehicle.displayDetailInfo();

        sc.close();
    }
}
```

OUTPUT:

```
PS C:\Users\karnish.n\Desktop\New folder> cd "c:\Users\karnish.n\Desktop\New folder\" ; if ($?
) { javac VehicleMain.java } ; if ($?) { java VehicleMain }
1. Two Wheeler
2. Four Wheeler
Enter Vehicle Type: 2
Vehicle Make: HONDA
Vehicle Number: TN 36 7379
Fuel Type (Petrol/Diesel): Petrol
Fuel Capacity: 10
CC: 120
Audio System: yes
Number of Doors: 4
*** HONDA ***
Vehicle Number: TN 36 7379
Fuel Type: Petrol
Vehicle Number: TN 36 7379
Fuel Type (Petrol/Diesel): Petrol
Fuel Capacity: 10
CC: 120
Audio System: yes
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.

CODING:

```java
class Shape {

    double calculateArea() {
        System.out.println("Calculating area of generic shape...");
        return 0;
    }
}



class Square extends Shape {
    double side;

    Square(double side) {
        this.side = side;
    }
}
```

```java
        double calculateArea() {
            return side * side;
        }
}


class Triangle extends Shape {
    double base, height;

    Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }



    double calculateArea() {
        return 0.5 * base * height;
    }
}


class Rectangle extends Shape {
    double length, width;

    Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }



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


public class Main {
    public static void main(String[] args) {

        Shape shape;
```

```
        shape = new Square(5);
        System.out.println("Area of Square: " + shape.calculateArea());

        shape = new Triangle(4, 6);
        System.out.println("Area of Triangle: " +
shape.calculateArea());

        shape = new Rectangle(7, 3);
        System.out.println("Area of Rectangle: " +
shape.calculateArea());
    }
}
```

OUTPUT:

```
Area of Square: 25.0
Area of Triangle: 12.0
Area of Rectangle: 21.0
```

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.

CODING:

Associate.java:

```java
public class Associate {
    private int associateId;
    private String associateName;
    private String workStatus;


    public Associate() {}

    public Associate(int associateId, String associateName, String
workStatus) {
        this.associateId = associateId;
        this.associateName = associateName;
        this.workStatus = 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 > 60) {
```

```java
            workStatus = "Deployed in project";
        } else if (days > 40) {
            workStatus = "Project phase";
        } else if (days > 20) {
            workStatus = "Advanced modules";
        } else if (days >= 0) {
            workStatus = "Core skills";
        } else {
            workStatus = "Invalid number of days";
        }
    }
}
```

Main.java:

```java
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        Associate associate = new Associate();

        System.out.print("Enter Associate ID: ");
        associate.setAssociateId(sc.nextInt());
        sc.nextLine();

        System.out.print("Enter Associate Name: ");
        associate.setAssociateName(sc.nextLine());

        System.out.print("Enter number of days completed in training: ");
        int days = sc.nextInt();

        associate.trackAssociateStatus(days);


        System.out.println("\nAssociate Details:");
        System.out.println("ID: " + associate.getAssociateId());
```

```
        System.out.println("Name: " + associate.getAssociateName());
        System.out.println("Work Status: " +
associate.getWorkStatus());


        sc.close();
    }
}
```

OUTPUT:

```
PS C:\Users\karnish.n\Desktop\New folder> cd "c:\Users\karnish.n\Desktop\New folder\task4\" ;
if ($?) { javac Main.java } ; if ($?) { java Main }
Enter Associate ID: 1001
Enter Associate Name: karnish
Enter number of days completed in training: 45

Associate Details:
ID: 1001
Name: karnish
Work Status: Project phase
PS C:\Users\karnish.n\Desktop\New folder\task4>
```