

Day 10

Dharunya S

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 StudentTest {
```

```
    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;
    }
    @Override
    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);
    }
    @Override
    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; }
    @Override

```

```

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);
}
@Override
public boolean isPassed() {
    return grade > 70 && noOfPapersPublished >= 2;
}
}
public static void main(String[] args) {
    UGStudent ug = new UGStudent("Meena", "UG567", 19, 68.0, "Bangalore", "BBA",
"Finance");
    ug.display();
    System.out.println("Passed: " + ug.isPassed());
    System.out.println();
    PGStudent pg = new PGStudent("Karthik", "PG890", 25, 73.5, "Hyderabad", "Data
Science", 1);
    pg.display();
    System.out.println("Passed: " + pg.isPassed());
    System.out.println();
    Student s = new Student("Anjali", "ST999", 18, 49.5, "Trichy");
    s.display();
    System.out.println("Passed: " + s.isPassed());
}
}

```

Output:

UG Student Details:

Name: Meena

ID: UG567

Age: 19

Grade: 68.0

Address: Bangalore

Degree: BBA

Stream: Finance

Passed: false

PG Student Details:

Name: Karthik

ID: PG890

Age: 25
Grade: 73.5
Address: Hyderabad
Specialization: Data Science
No of Papers Published: 1
Passed: false
Student Details:
Name: Anjali
ID: ST999
Age: 18
Grade: 49.5
Address: Trichy
Passed: false

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 VehicleApp {
    static class Vehicle {
        protected String make;
        protected String vehicleNumber;
        protected String fuelType;
        protected Integer fuelCapacity;
        protected Integer cc;
        public Vehicle(String make, String vehicleNumber, String fuelType, Integer fuelCapacity,
Integer 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 Integer getFuelCapacity() { return fuelCapacity; }
        public void setFuelCapacity(Integer fuelCapacity) { this.fuelCapacity = fuelCapacity; }
        public Integer getCc() { return cc; }
        public void setCc(Integer 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("Engine CC: " + cc);
        }
        public void displayDetailInfo() {
        }
    }
}
static class TwoWheeler extends Vehicle {
    private boolean kickStartAvailable;
```

```

        public TwoWheeler(String make, String vehicleNumber, String fuelType, Integer
fuelCapacity, Integer 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"));
        }
    }
    // FourWheeler class
    static class FourWheeler extends Vehicle {
        private String audioSystem;
        private int numberOfDoors;
        public FourWheeler(String make, String vehicleNumber, String fuelType, Integer
fuelCapacity, Integer 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; }
        @Override
        public void displayDetailInfo() {
            System.out.println("Audio System: " + audioSystem);
            System.out.println("Number of Doors: " + numberOfDoors);
        }
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("1. Two Wheeler");
        System.out.println("2. Four Wheeler");
        System.out.print("Enter Vehicle Type (1 or 2): ");
        int choice = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Vehicle Make: ");
        String make = scanner.nextLine();
        System.out.print("Enter Vehicle Number: ");

```

```

String vehicleNumber = scanner.nextLine();
System.out.print("Enter Fuel Type (Petrol/Diesel): ");
String fuelType = scanner.nextLine();
System.out.print("Enter Fuel Capacity: ");
int fuelCapacity = scanner.nextInt();
System.out.print("Enter Engine CC: ");
int cc = scanner.nextInt();
Vehicle vehicle;
if (choice == 1) {
    scanner.nextLine();
    System.out.print("Kick Start Available (yes/no): ");
    String kickStart = scanner.nextLine();
    boolean isKickStart = kickStart.equalsIgnoreCase("yes");
    vehicle = new TwoWheeler(make, vehicleNumber, fuelType, fuelCapacity, cc,
isKickStart);
} else if (choice == 2) {
    scanner.nextLine();
    System.out.print("Enter Audio System: ");
    String audioSystem = scanner.nextLine();
    System.out.print("Enter Number of Doors: ");
    int numberOfDoors = scanner.nextInt();
    vehicle = new FourWheeler(make, vehicleNumber, fuelType, fuelCapacity, cc,
audioSystem, numberOfDoors);
} else {
    System.out.println("Invalid choice!");
    scanner.close();
    return;
}
System.out.println("Vehicle Information");
vehicle.displayMake();
vehicle.displayBasicInfo();
vehicle.displayDetailInfo();
scanner.close();
}
}

```

output:

```

1. Two Wheeler
2. Four Wheeler
Enter Vehicle Type (1 or 2): 1
Enter Vehicle Make: yamaha
Enter Vehicle Number: 123
Enter Fuel Type (Petrol/Diesel): petrol
Enter Fuel Capacity: 60
Enter Engine CC: 300

```


Kick Start Available (yes/no): yes
Vehicle Information
Vehicle Make: yamaha
Vehicle Number: 123
Fuel Type: petrol
Fuel Capacity: 60
Engine CC: 300
Kick Start Available: Yes

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;
class Shape {
    double calculateArea() {
        return 0;
    }
}
class Square extends Shape {
    double side;
    Square(double side) {
        this.side = side;
    }
    @Override
    double calculateArea() {
        return side * side;
    }
}
class Rectangle extends Shape {
    double length, width;
    Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    @Override
    double calculateArea() {
        return length * width;
    }
}
class Triangle extends Shape {
    double base, height;
    Triangle(double base, double height) {
```

```

        this.base = base;
        this.height = height;
    }
    @Override
    double calculateArea() {
        return 0.5 * base * height;
    }
}

public class ShapeApp {
    public static void main(String[] args) {
        Shape s;
        s = new Square(5);
        System.out.println("Area of Square: " + s.calculateArea());
        s = new Rectangle(4, 6);
        System.out.println("Area of Rectangle: " + s.calculateArea());
        s = new Triangle(3, 7);
        System.out.println("Area of Triangle: " + s.calculateArea());
    }
}

```

output:

```

Area of Square: 25.0
Area of Rectangle: 24.0
Area of Triangle: 10.5

```

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 yet 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 days) {
    if (days > 60) {
        this.workStatus = "Deployed in project";
    } else if (days > 40) {
        this.workStatus = "Project phase";
    } else if (days > 20) {
        this.workStatus = "Advanced modules";
    } else if (days > 0) {
        this.workStatus = "Core skills";
    } else {
        this.workStatus = "Training not started";
    }
}
}

public class Associates {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Associate ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
    }
}

```

```
System.out.print("Enter Associate Name: ");
String name = scanner.nextLine();
System.out.print("Enter number of days completed: ");
int days = scanner.nextInt();
Associate associate = new Associate(id, name);
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());
scanner.close();
}
}
```

Output:

Enter Associate ID: 10

Enter Associate Name: anu

Enter number of days completed: 50

Associate Details:

ID: 10

Name: anu

Work Status: Project phase