Ex no. :3 Name: M.Rohith
Date: 18-10-2022 Reg.no. : 3122 21 5001 085

SN College of Engineering

Department of Computer Science and Engineering

UCS2313 – Object Oriented Programming Lab

<u>II Year CSE - B Section (III Semester)</u>

Academic Year 2022-23

Batch: 2021- 2025

Faculty Incharge :S.Rajalakshmi

<u>Exercise 3. Inheritance</u>

<u>Objective:</u>

1. To test the following Inheritance types: single-level, multi-level and hierarchical inheritance.
2. To test the scope of private and protected variables, constructors in inherited class hierarchy.

<u>Sample Learning Outcome:</u>

1. Need of inheritance and its implementation in Java
2. Type of inheritance
3. Working of constructors in inherited class
4. Accessing inherited class through base class reference
5. Method overloading and overriding in inheritance

<u>Best Practices:</u>

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

I) Create a class hierarchy for the classes defined below: Design a class called Person as described below:

-	Private members
+	Public members
#	Protected members
~	Default (Package private)

Person
-aadhaar:int -name:String -address:String -gender:char
+Person(aadhaar,name,address,gender) +getName():String +getAddress():String +setAddress(address):void +getGender():char

A sub-class Student of class Person is designed as shown below:

Student
-program:String -year:int -totalmark:float
+Student(aadhaar,name,address,gender,program,year,total) +getProgram():String +getYear():int +setYear(year):void +getTotal():float +setTotal(tot):void +calGPA():float

A sub-class Faculty of class Person is designed as shown below:

Faculty
-designation:String -department:String -basicpay:float
+Faculty(aadhaar,name,address,gender,designation,dept,pay)

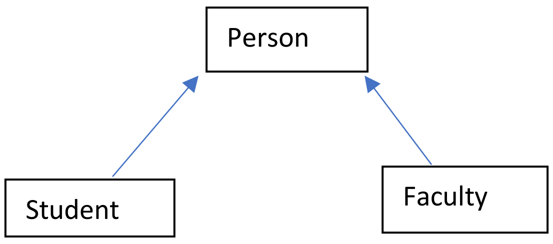
+getDesig():String +setDesig(desig):void +setBasic(basic):void +getBasic():float +calSalary():float

Note the following:

1. The hierarchy Person -> Student (or) Person -> Faculty is a ***Single-level*** *inheritance* type.
2. The type of above entire class hierarchy (Person -> Student , Person -> Faculty) is the ***Hierarchical*** *Inheritance*.
3. Note the use of constructors at all levels of class hierarchy.

EXERCISE : I)

1.Draw the class diagram of the above class hierarchy.



2. Write a *test driver* called `TestInheritance` to test all the `public` methods that display the student and faculty details.

Use the following to calculate Net Salary:

Gross salary = Basicpay + DA as 60% of basic + HRA as 10% of basic

Deductions = Medical Insurance as 8.5% of basic + PF as 8% of basic

Net salary = Gross salary – Deductions

Ex no. :3 Name: M.Rohith
Date: 18-10-2022 Reg.no. : 3122 21 5001 085

Program code:

```java
import java.util.*;
class Person
{
 // Class Members
    private long aadhar;
    private String name, address;
    private char gender;
 // Empty Constructor
    public Person(){
    }
 // Arguement Constructor
    public Person(long aadhar, String name, String address, char gender)
    {
        this.aadhar = aadhar;
        this.name = name;
        this.address = address;
        this.gender = gender;
    }
    // Setters
    public void setAadhar(long aadhar)
    {
        this.aadhar = aadhar;
    }
    public void setName(String name)
    {
        this.name = name;
    }
    public void setAddress(String address)
    {
        this.address = address;
    }
    public void setGender(char gender)
    {
        this.gender = gender;
    }
    // Getters
    public long getAadhar()
    {
        return this.aadhar;
    }
    public String getName()
    {
        return this.name;
    }
    public String getAddress()
    {
        return this.address;
    }
    public char getGender()
    {
        return this.gender;
    }
}
class Student extends Person
{
    // Class Members
    private String program;
    private int year;
    private float totalmark;
    private int totalsem;

    // Empty Constructor
    public Student()
    {
    }
    // Arguement Constructor
    public Student(long aadhar, String name, String address, char gender,
String program, int year, float totalmark, int totalsem)
    {
        super(aadhar, name, address, gender);
        this.program = program;
        this.year = year;
        this.totalmark = totalmark;
        this.totalsem = totalsem;
    }
    // Setters
    public void setProgram(String program)
    {
        this.program = program;
    }
    public void setYear(int year)
    {
        this.year = year;
    }
    public void setTotalMark(float totalmark)
    {
        this.totalmark = totalmark;
    }
    public void setTotalSem(int totalsem)
    {
        this.totalsem = totalsem;
    }
    // Getters
    public String getProgram()
    {
        return this.program;
    }
    public int getYear()
    {
        return this.year;
    }
    public float getTotalMark()
    {
        return this.totalmark;
    }
    public float getTotalSem()
    {
        return this.totalsem;
    }
    // Calculate GPA
    public float calGPA()
    {
        return (float) (this.getTotalMark() / this.getTotalSem());
    }
}
class Faculty extends Person
{
    // Class Members
    private String designation, department;
    private float basicpay;
    // Empty Constructor
    public Faculty()
    {
    }
    // Arguement Constructor
    public Faculty(long aadhar, String name, String address, char gender, 
String designation, String department, float basicpay)
    {
        super(aadhar, name, address, gender);
        this.designation = designation;
        this.department = department;
        this.basicpay = basicpay;
    }
    // Setters
    public void setDesignation(String designation)
    {
        this.designation = designation;
    }
    public void setDepartment(String department)
    {
        this.department = department;
    }
    public void setBasicPay(float basicpay)
    {
        this.basicpay = basicpay;
    }
    // Getters
    public String getDesignation()
    {
        return this.designation;
    }
    public String getDepartment()
    {
        return this.department;
    }
    public float getBasicPay()
    {
        return this.basicpay;
    }
    // Calculte Salary
    public float calSalary()
    {
        return (float) (((0.6 * getBasicPay()) + (0.1 * getBasicPay())) -
((0.085 * getBasicPay()) + (0.08 * getBasicPay())));
    }
}
class TestInheritance{
    public static void main(String[] args)
    {
        Student student1 = new Student();
        setDetails(student1);
        displayDetails(student1);
        Faculty faculty1 = new Faculty(30_232_1323, "Karthi", "No. 42, Ganesh
Nagar, II Street,Tuticorin-8.", 'M',"Professor", "Computer Science and
Engineering", (float)20000.00);
        displayDetails(faculty1);
    }
    public static void displayDetails(Student student)
    {
        System.out.println("\nThe Student's name is: " +
        student.getName());
        System.out.println("The Student's gender is: " +
        student.getGender());
        System.out.println("The Student's aadhar number is: " +
        student.getAadhar());
        System.out.println("The Student's address is: " +
        student.getAddress());
        System.out.println("The Student's opted program is: " +
        student.getProgram());
        System.out.println("The Student's current year is: " +
```

```java
        student.getYear());
        System.out.println("The Student's CGPA is: " +
        student.calGPA());
    }
    public static void displayDetails(Faculty faculty)
    {
        System.out.println("\nThe Faculty's name is: " +
        faculty.getName());
        System.out.println("The Faculty's gender is: " +
        faculty.getGender());
        System.out.println("The Faculty's aadhar number is: " +
        faculty.getAadhar());
        System.out.println("The Faculty's address is: " +
        faculty.getAddress());
        System.out.println("The Faculty's designation is: " +
        faculty.getDesignation());
        System.out.println("The Faculty's department is: " +
        faculty.getDepartment());
        System.out.println("The Faculty's basicpay is: " +
        faculty.getBasicPay());
    }
    public static void setDetails(Student student)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("\nEnter Student's name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Student's gender: ");
        char gender = scanner.next().charAt(0);
        System.out.print("Enter Student's aadhar: ");
        long aadhar = scanner.nextLong();
        System.out.print("Enter Student's address: ");
        String address = scanner.nextLine();
        scanner.nextLine();
        System.out.print("Enter Student's program: ");
        String program = scanner.nextLine();
        System.out.print("Enter Student's current year: ");
        int year = scanner.nextInt();
        System.out.print("Enter Student's total GPA: ");
        float GPA = scanner.nextFloat();
        System.out.print("Enter number of semesters attended by student: ");
        int totalsem = scanner.nextInt();
        student.setName(name);
        student.setGender(gender);
        student.setAadhar(aadhar);
        student.setAddress(address);
        student.setProgram(program);
        student.setYear(year);
        student.setTotalMark(GPA);
        student.setTotalSem(totalsem);
    }
}
```

Output:

```
Enter Student's name: Ajith
Enter Student's gender: M
Enter Student's aadhar: 123242423
Enter Student's address: 72A/1/5 ABC Colony,Millerpuram,Tuticorin-8
Enter Student's program: Computer Science and Engineering
Enter Student's current year: 2
Enter Student's total GPA: 20
Enter number of semesters attended by student: 2

The Student's name is: Ajith
The Student's gender is: M
The Student's aadhar number is: 123242423
The Student's address is:
The Student's opted program is: Computer Science and Engineering
The Student's current year is: 2
The Student's CGPA is: 10.0

The Faculty's name is: Karthi
The Faculty's gender is: M
The Faculty's aadhar number is: 302321323
The Faculty's address is: No. 42, Ganesh Nagar, II Street,Tuticorin-8.
The Faculty's designation is: Professor
The Faculty's department is: Computer Science and Engineering
The Faculty's basicpay is: 20000.0
```

%%%%%%%%%%%%%%%%%%%##########%%%%%%%%%%%%%%%%%######

Ex no. :3 Name: M.Rohith
Date: 18-10-2022 Reg.no. : 3122 21 5001 085

II) Create a class hierarchy for the classes/interface as defined below: Design a class Shape as described below: # - *protected*

Shape
#color:String=”red”

+Shape()
+Shape(color)
+getColor():String
+setColor(color):void

A sub-class **Circle** of class *Shape* is designed as shown below:

Circle
#radius:float=1.0
+Circle() +Circle(radius) +Circle(radius,color) +getRadius():float +setRadius(radius):void +getArea():float +getPerimeter():float

A sub-class **Rectangle** of class *Shape* is designed as shown below:

Rectangle
#width:float=1.0 #length:float=1.0
+Rectangle() +Rectangle(width,length) +Rectangle(width,length,color) +getWidth():float +setWidth(width):void +getLength():float +setLength(length):void +getArea():float +getPerimeter():float

Ex no. :3 Name: M.Rohith
Date: 18-10-2022 Reg.no. : 3122 21 5001 085

A sub-class **Square** of class *Rectangle* is designed as shown below:

Square

+Square()
+Square(side)
+Square(side,color)
+getSide():float
+setSide(side):void

Note the following:

1. The hierarchy Shape --> Rectangle --> Square is a ***Multi-level*** *inheritance* type.

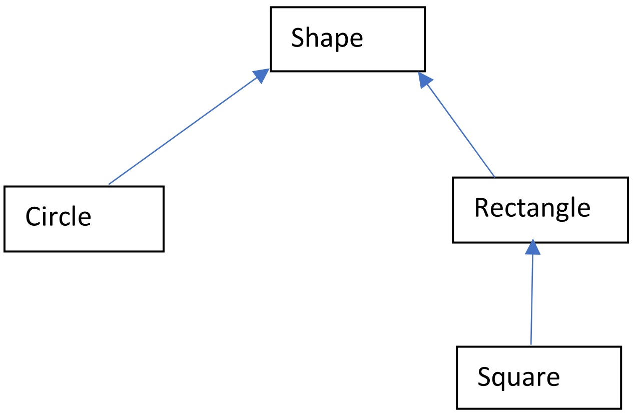
2. The type of above entire class hierarchy is the ***Hierarchical*** *Inheritance*.

3. Note the constructor overloading at all the levels.

4. # denotes `protected` variable. The `protected` variables can be accessed by its subclasses and classes in the same package.

EXERCISE : II)

1.Draw the class diagram of the above class hierarchy.



Ex no. :3 Name: M.Rohith
Date: 18-10-2022 Reg.no. : 3122 21 5001 085

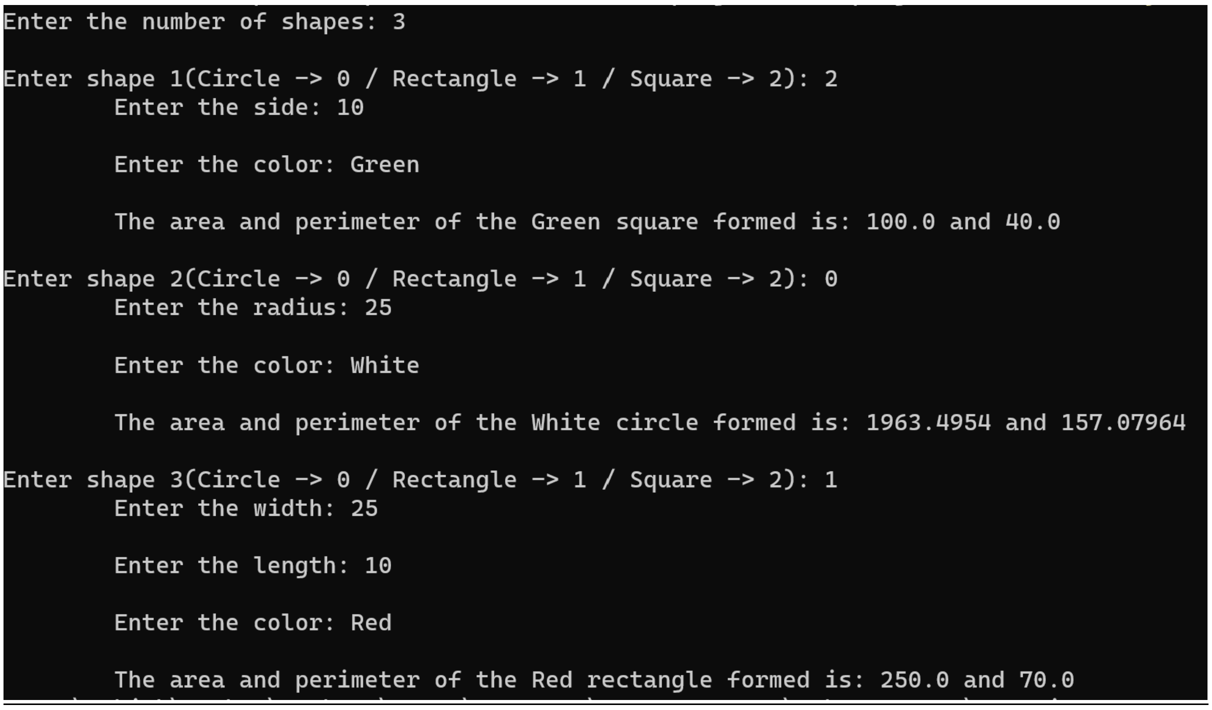
2. Write a *test driver* called `TestShape` to test all the `public` methods. Use an array of objects of type *Shape* and display the area and perimeter of all the shapes (Circle, Rectangle and Square).

<u>Program code:</u>

```java
import java.util.*;
class Shape
{
    // Class Members
    protected String color = "red";
    // Default Constructor
    public Shape()
    {
    }
    // Arguement Constructor
    public Shape(String color)
    {
        this.color = color;
    }
    // Setter
    public void setColor(String color)
    {
        this.color = color;
    }
    // Getter
    public String getColor()
    {
        return this.color;
    }
    public float getArea()
    {
        return 0;
    }
    public float getPerimeter()
    {
        return 0;
    }
}
class Circle extends Shape
{
    // Class Members
    protected float radius = 1.0f;
    // Default Constructor
    public Circle()
    {
    }
    // Arguement Constructors
    public Circle(float radius)
    {
        this.radius = radius;
    }
    public Circle(float radius, String color)
    {
        super(color);
        this.radius = radius;
    }
    // Setter
    public void setRadius(float radius)
    {
        this.radius = radius;
    }
    // Getter
    public float getRadius()
    {
        return this.radius;
    }
    @Override
    public float getArea()
    {
        return (float) (Math.PI * this.getRadius() * this.getRadius());
    }
    @Override
    public float getPerimeter()
    {
        return (float) (2 * Math.PI * this.getRadius());
    }
}
class Rectangle extends Shape
{
    // Class Members
    protected float width = 1.0f;
    protected float length = 1.0f;
    // Default Constructor
    public Rectangle()
    {
    }
    // Arguement Constructors
    public Rectangle(float width, float length)
    {
        this.width = width;
        this.length = length;
    }
    public Rectangle(float width, float length, String color)
    {
        super(color);
        this.width = width;
        this.length = length;
    }
    // Setters
    public void setWidth(float width)
    {
        this.width = width;
    }
    public void setLength(float length)
    {
        this.length = length;
    }
    // Getters
    public float getWidth()
    {
        return this.width;
    }
    public float getLength()
    {
        return this.length;
    }
    @Override
    public float getArea()
    {
        return (float) (this.getLength() * this.getWidth());
    }
    @Override
    public float getPerimeter()
    {
        return (float) (2 * (this.getLength() + this.getWidth()));
    }
}
class Square extends Rectangle
{
    // No Class Members
    // Default Constructor
    public Square()
    {
    }
    // Arguement Constructors
    public Square(float side)
    {
        super(side, side);
    }
```

```java
    public Square(float side, String color)
    {
        super(side, side, color);
    }
    // Setter
    public void setSide(float side)
    {
        super.setWidth(side);
        super.setLength(side);
    }
    public float getSide()
    {
        return super.getWidth();
    }
}
class TestShape
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of shapes: ");
        int n = scanner.nextInt();
        Shape[] shapes = new Shape[n];
        for (int i = 0; i < n; i++)
        {
            System.out.print("\nEnter shape " + (i + 1) + "(Circle -> 0 / 
Rectangle -> 1 / Square -> 2): ");
            int shapeName = scanner.nextInt();
            switch(shapeName){
            case 0:
                System.out.print("\tEnter the radius: ");
                float radius = scanner.nextFloat();
                System.out.print("\n\tEnter the color: ");
                String colorCir = scanner.next();
                shapes[i] = new Circle(radius, colorCir);
                System.out.println("\n\tThe area and perimeter of the " + 
shapes[i].getColor() + " circle formed is: " + shapes[i].getArea() + " and " + 
shapes[i].getPerimeter());
                break;
            case 1:
                System.out.print("\tEnter the width: ");
                float width = scanner.nextFloat();
                System.out.print("\n\tEnter the length: ");
                float length = scanner.nextFloat();
                System.out.print("\n\tEnter the color: ");
                String colorRect = scanner.next();
                shapes[i] = new Rectangle(width, length, colorRect);
                System.out.println("\n\tThe area and perimeter of the " + 
shapes[i].getColor() + " rectangle formed is: " + shapes[i].getArea() + " and 
" + shapes[i].getPerimeter());
                break;
            case 2:
                System.out.print("\tEnter the side: ");
                float side = scanner.nextFloat();
                System.out.print("\n\tEnter the color: ");
                String colorSq = scanner.next();
                shapes[i] = new Square(side, colorSq);
                System.out.println("\n\tThe area and perimeter of the " + 
shapes[i].getColor() + " square formed is: " + shapes[i].getArea() + " and " + 
shapes[i].getPerimeter());
                break;
            }
        }
    }
}
```

Output:



```
Enter the number of shapes: 3

Enter shape 1(Circle -> 0 / Rectangle -> 1 / Square -> 2): 2
        Enter the side: 10

        Enter the color: Green

        The area and perimeter of the Green square formed is: 100.0 and 40.0

Enter shape 2(Circle -> 0 / Rectangle -> 1 / Square -> 2): 0
        Enter the radius: 25

        Enter the color: White

        The area and perimeter of the White circle formed is: 1963.4954 and 157.07964

Enter shape 3(Circle -> 0 / Rectangle -> 1 / Square -> 2): 1
        Enter the width: 25

        Enter the length: 10

        Enter the color: Red

        The area and perimeter of the Red rectangle formed is: 250.0 and 70.0
```

Ex no. :3 Name: M.Rohith

Date: 18-10-2022 Reg.no. : 3122 21 5001 085

Learning outcomes:

Java program to test the Inheritance types: single-level, multi-level and hierarchical inheritance has been implemented and executed and also the scope of private and protected variables, constructors in inherited class hierarchy has been tested and implemented in the given programs.

#############$$$$$$$$$$$$$$$$$$##################$$$$$$$$$$$$$$$####