

ASSIGNMENT – 5

1. Write a program that creates a class Account that have members customer name, account number. Account has constructor to initialize its members and method display() to show the result. Create a child class Savings_Account that is derived from Account class. Savings_Account have members min_bal and saving_bal. Use show() to display its details. Then create another child class Account_details from Savings_Account class have members deposit, withdrawl and a method show1() to show its details. Child classes have constructors of their own. Create a driver class that creates a record of customer and display all its details.

```
        public void main(String[] args)

AccountDetails customer = new AccountDetails(

    "Naren Sahoo",

    9974573201L,

    2000.0,

    8000.0,

    3500.0,

    1000.0

);

System.out.println("----- CUSTOMER ACCOUNT DETAILS -----\\n");

customer.displayAccountInfo();

customer.displaySavingsInfo();

customer.displayFullDetails();

}

class Account {

    protected String customerName;

    protected long accountNumber;

    public Account(String customerName, long accountNumber) {

        this.customerName = customerName;

        this.accountNumber = accountNumber;

        public void displayAccountInfo() {

            System.out.println("Customer Name : " + customerName);

            System.out.println("Account Number : " + accountNumber);

        }

    }

}

class SavingsAccount extends Account {

    protected double minBalance;
```

```

    protected double savingsBalance;

    public SavingsAccount(String customerName, long accountNumber, double minBalance, double
savingsBalance) {

        super(customerName, accountNumber);

        this.minBalance = minBalance;

        this.savingsBalance = savingsBalance;
    }

    public void displaySavingsInfo() {

        System.out.println("Minimum Balance : " + minBalance);

        System.out.println("Saving Balance : " + savingsBalance);

    }
}

class AccountDetails extends SavingsAccount {

    private double depositAmount;

    private double withdrawalAmount;

    public AccountDetails(String customerName, long accountNumber,

        double minBalance, double savingsBalance,

        double depositAmount, double withdrawalAmount) {

        super(customerName, accountNumber, minBalance, savingsBalance);

        this.depositAmount = depositAmount;

        this.withdrawalAmount = withdrawalAmount;

        this.savingsBalance += depositAmount;

        this.savingsBalance -= withdrawalAmount;

    }

    public void displayFullDetails() {

        System.out.println("Deposit Amount : " + depositAmount);

        System.out.println("Withdrawal Amt : " + withdrawalAmount);

        System.out.println("Updated Balance : " + savingsBalance);

    }
}

```

OUTPUT :

```

C:\Users\Padmalya Meher\Desktop\25bcs148>java Account.java
----- CUSTOMER ACCOUNT DETAILS -----
Customer Name   : Naren Sahoo
Account Number  : 9974573201
Minimum Balance : 2000.0
Saving Balance  : 10500.0
Deposit Amount  : 3500.0
Withdrawal Amt  : 1000.0
Updated Balance : 10500.0

```

2. Create a class Figure with instance members dim1 and dim2. Use constructor and area() that returns the area of figure. Create a derived class Rectangle derived from Figure and area() that returns the area of rectangle. Create another derived class Triangle and Square that has area() which returns area of Triangle and Square respectively. Derived class have appropriate constructor. Using method overriding concept test the functionalities of derived class by creating objects and super class memory references in Driver class.

```
class Figure {
    protected double dim1;
    protected double dim2;
    Figure(double dim1, double dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }
    double area() {
        return 0;
    }
}

class Rectangle extends Figure {
    Rectangle(double length, double breadth) {
        super(length, breadth);
    }
    double area() {
        return dim1 * dim2;
    }
}

class Triangle extends Figure {
    Triangle(double base, double height) {
        super(base, height);
    }
    double area() {
        return 0.5 * dim1 * dim2;
    }
}

class Square extends Figure {
    Square(double side) {
        super(side, side); // both dim1 and dim2 will be side
    }
    double area() {
        return dim1 * dim1; // or dim1 * dim2 (both same)
    }
}
```

```

    }
}

public void main(String[] args) {
    Figure fig;
    fig = new Rectangle(10, 5);
    System.out.println("Area of Rectangle: " + fig.area());
    fig = new Triangle(10, 5);
    System.out.println("Area of Triangle: " + fig.area());
    fig = new Square(6);
    System.out.println("Area of Square: " + fig.area());
}

```

OUTPUT :

```

C:\Users\Padmalya Meher\Desktop\25bcs148>javac Figure.java
C:\Users\Padmalya Meher\Desktop\25bcs148>java Figure.java
Area of Rectangle: 50.0
Area of Triangle: 25.0
Area of Square: 36.0

```

3.Write a program to create a class named Shape. It should contain two methods, draw() and erase() that prints "Drawing Shape" and "Erasing Shape" respectively. For this class, create three sub classes, Circle, Triangle and Square and each class should override the parent class functions - draw () and erase (). The draw() method should print "Drawing Circle", "Drawing Triangle" and "Drawing Square" respectively. The erase() method should print "Erasing Circle", "Erasing Triangle" and "Erasing Square" respectively. Create objects of Circle, Triangle and Square, assign each to Shape variable(reference) and call draw() and erase() method using each object.

```

class Shape {
    void draw() {
        System.out.println("Drawing Shape");
    }
    void erase() {
        System.out.println("Erasing Shape");
    }
}

class Circle extends Shape {
    void draw() {
        System.out.println("Drawing Circle");
    }
    void erase() {
        System.out.println("Erasing Circle");
    }
}

```

```

class Triangle extends Shape {
    void draw() {
        System.out.println("Drawing Triangle");
    }
    void erase() {
        System.out.println("Erasing Triangle");
    }
}
class Square extends Shape {
    void draw() {
        System.out.println("Drawing Square");
    }
    void erase() {
        System.out.println("Erasing Square");
    }
}

public void main(String[] args) {
    Shape shape;
    shape = new Circle();
        shape.draw();
        shape.erase();
    shape = new Triangle();
        shape.draw();
        shape.erase();
    shape = new Square();
        shape.draw();shape.erase();
}

```

OUTPUT:

```

C:\Users\Padmalya Meher\Desktop\25bcs148>javac Shape.java
C:\Users\Padmalya Meher\Desktop\25bcs148>java Shape.java
Drawing Circle
Erasing Circle
Drawing Triangle
Erasing Triangle
Drawing Square
Erasing Square

```

4. Define an abstract class named "Figure", having data members dim1 and dim2. Extend this class to create two concrete classes named Rectangle and Triangle. Override the getArea() method in the sub classes. Invoke the getArea() method in the main method of another Driver class through the abstract class reference variable.

```

abstract class Figure {
    double dim1, dim2;
    Figure(double dim1, double dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }
    abstract double getArea(); // Abstract method
}

class Rectangle extends Figure {
    Rectangle(double length, double breadth) {
        super(length, breadth);
    }
    double getArea() {
        return dim1 * dim2;
    }
}

class Triangle extends Figure {
    Triangle(double base, double height) {
        super(base, height);
    }
    double getArea() {
        return 0.5 * dim1 * dim2;
    }
}

public void main(String[] args) {
    Figure figure;
    figure = new Rectangle(10, 5);
    System.out.println("Area of Rectangle: " + figure.getArea());
    figure = new Triangle(10, 5);
    System.out.println("Area of Triangle: " + figure.getArea());
}

```

OUTPUT :

```

C:\Users\Padmalya Meher\Desktop\25bcs148>java Figure.java
Area of Rectangle: 50.0
Area of Triangle: 25.0
C:\Users\Padmalya Meher\Desktop\25bcs148>

```

5. Create a class Point2D with the data member x and y coordinate. Use default and parameterised constructor to set the coordinate values and display() to show the coordinates. Create a subclass called Point3D which is derived from the superclass Point2D with data members z coordinate and has

constructor to initialize the input and show() method to display the coordinates. Test the methods of both the classes by creating objects in the main method of driver class.

```
public void main(String[] args) {  
    Point2D pointA = new Point2D();  
    Point2D pointB = new Point2D(4, 7);  
    pointA.display();  
    pointB.display();  
    Point3D pointC = new Point3D(1, 2, 3);  
    pointC.show();  
}  
  
class Point2D {  
    protected int x;  
    protected int y;  
    public Point2D() {  
        this.x = 0;  
        this.y = 0;  
    }  
    public Point2D(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
    public void display() {  
        System.out.println("Point2D Coordinates: (" + x + ", " + y + ")");  
    }  
}  
  
class Point3D extends Point2D  
{  
    private int z;  
    public Point3D(int x, int y, int z) {  
        super(x, y); // Call superclass constructor  
        this.z = z;  
    }  
    public void show() {  
        System.out.println("Point3D Coordinates: (" + x + ", " + y + ", " + z + ")");  
    }  
}
```

OUTPUT :

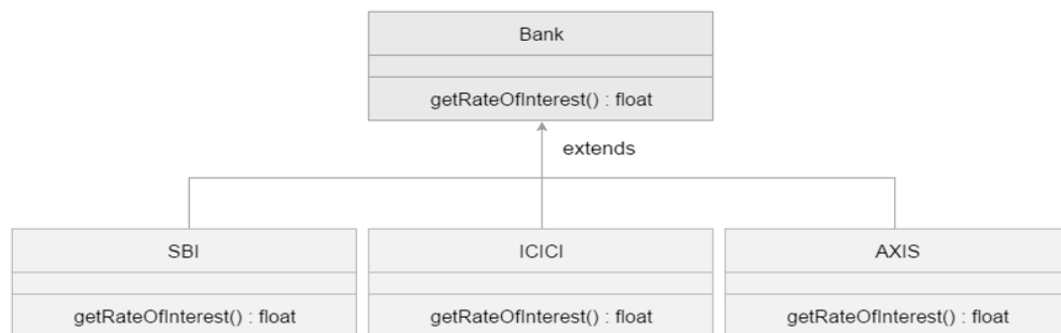
```
2 errors
C:\Users\Padmalya Meher\Desktop\25bcs148>javac Point2D.java
C:\Users\Padmalya Meher\Desktop\25bcs148>java Point2D.java
Point2D Coordinates: (0, 0)
Point2D Coordinates: (4, 7)
Point3D Coordinates: (1, 2, 3)
```

6. Create the classes as given in the below figure. Display the interest rate in the following format:

SBI Rate of Interest : 8

ICICI Rate of Interest : 7

AXIS Rate of Interest : 9



```
class Bank {
    float getRateOfInterest() {
        return 0;
    }
}
class SBI extends Bank {
    float getRateOfInterest() {
        return 8.0f;
    }
}
class ICICI extends Bank {
    float getRateOfInterest() {
        return 7.0f;
    }
}
class AXIS extends Bank {
    float getRateOfInterest() {
        return 9.0f;
    }
}
public void main(String[] args) {
    Bank sbi = new SBI();
    Bank icici = new ICICI();
}
```



```

    Bank axis = new AXIS();

    System.out.println("SBI Rate of Interest : " + sbi.getRateOfInterest());

    System.out.println("ICICI Rate of Interest : " + icici.getRateOfInterest());

    System.out.println("AXIS Rate of Interest : " + axis.getRateOfInterest());

}

```

OUTPUT:

```

C:\Users\Padmalya Meher\Desktop\25bcs148>javac Bank.java
C:\Users\Padmalya Meher\Desktop\25bcs148>java Bank.java
SBI Rate of Interest : 8.0
ICICI Rate of Interest : 7.0
AXIS Rate of Interest : 9.0

```

7. Create a class Person that has data member name. Use constructor to initialize name and display() to display name. Create a derived class Employee from Person class having private members empid. Using constructor initialize empid and have method display() to display empid. Create another derived class HourlyEmployee from Employee with private members hourlyRate and hoursWorked. Use constructor to initialize input and methods getGrossPay() that computes and returns the gross pay of the employee and display() to display the hourlyRate, hoursWorked and gross pay. Create a driver class to test the functionalities of the above classes and display output in the following format

Name : John Smith

EmpID : 7569

Hourly Rate : 100

Hours worked : 2000

Gross pay : 200000

```

public void main(String[] args) {
    HourlyEmployee emp = new HourlyEmployee("Jhon Smith", 7569, 100, 2000);
    emp.display();
}

class Person {
    protected String name;

    public Person(String name) {
        this.name = name;
    }

    public void display() {
        System.out.println("Name : " + name);
    }
}

class Employee extends Person {
    private int empid;

    public Employee(String name, int empid) {

```

```

        super(name);
        this.empid = empid;
    }
    public void display() {
        super.display(); // Call Person's display
        System.out.println("EmpID : " + empid);
    }
}

class HourlyEmployee extends Employee {
    private double hourlyRate;
    private int hoursWorked;

    public HourlyEmployee(String name, int empid, double hourlyRate, int hoursWorked) {
        super(name, empid);
        this.hourlyRate = hourlyRate;
        this.hoursWorked = hoursWorked;
    }

    public double getGrossPay() {
        return hourlyRate * hoursWorked;
    }

    public void display() {
        super.display();

        System.out.println("Hourly Rate : " + hourlyRate);
        System.out.println("Hours worked : " + hoursWorked);
        System.out.println("Gross pay : " + getGrossPay());
    }
}

```

OUTPUT :

```

C:\Users\Padmalya Meher\Desktop\25bcs148>javac Person.java
C:\Users\Padmalya Meher\Desktop\25bcs148>java Person.java
Name : Jhon Smith
EmpID : 7569
Hourly Rate : 100.0
Hours worked : 2000
Gross pay : 200000.0

```

Name : Padmalaya Meher

Sic :25bcs148 Roll.no : 01

Date of Expt :19/09/25