

S12BLH31

PROGRAMMING IN JAVA

Unit – II OBJECT ORIENTED PROGRAMMING

2a. IMPLEMENTATION OF PARAMETERIZED CONSTRUCTOR

Define a class Employee_Sal : Class name : Employee_Sal

Data members/Instance variables:

String name : to store name of the

employeeString empno : to store employee
number

int basic :: to store basic salary of the

employeeMember Methods:

i. A parameterised constructor to initialize the data members

ii. To accept the details of an employee

iii. To compute the gross and net salary

as:da = 30% of basic

hra = 15% of

basicpf =12%of

basic

gross =basic + da + hra

net = gross - pf

iv. To display the name, empno, gross salary, net salary.

Write a main method to create an object of a class and call the above member methods

// To calculate the gross and net salary of an employee

```
import java.util.*;
```

```
{
```

```
public class Employee_Sal
```

```
{
```

```
String name, empno;
```

```
int basic;
```

```
double da,hra,pf,gs,net;
```

```
Employee_Sal (String n, String en, int bs)
```

```
{
```

```
    name=n;
```

```
    empno=en;
```

```
    basic=bs;
```

```
}
```

```
void compute()
```

```
{
```

```
    da= basic*30.0/100.0;
```

```
    hra=basic*15.0/100.0;
```

```
    pf=basic*12.0/100.0;
```

```
    gs=basic+da+hra; net=gs-pf;
```

```
}
```

```

void display()
{
    System.out.println("Name:" + name);
    System.out.println("Employee Number :"+ empno);
    System.out.println("Gross salary : Rs. "+gs);
    System.out.println("Net Salary : Rs. "+net);
}

public static void main(String args[])
{
    Scanner in = new Scanner(System.in);
    String nm,enm;
    int bsal;
    System.out.println("Enter Employee's Name, Employee No, Basic salary :");
    nm=in.nextLine();
    enm=in.next();
    bsal=in.nextInt();
    Employee_Sal ob=new
    Employee_Sal(nm,enm,bsal);ob.compute();
    ob.display();
}
}

```

Output:

```

Enter Employee's Name, Employee No, Basic
salary :Madhavan
TS/10
1
32000

```

```

Name: Madhavan
Employee Number : TS/101
Gross salary : Rs. 46400.0
Net Salary : Rs. 42560.0

```

2b. CREATING A PROGRAM TO INITIALIZE THE CONSTRUCTOR

The 'Cabservice' is an organisation that provides 'Online Booking' for the passengers to avail pick-up and drop facility. Define a class Cabservice having the following specifications:

Class name : Cabservice

Instance variables/Data members:

String taxino : to store taxi
number

String name : to store name of the
passenger
int d : to store the distance
travelled (in km)

Member Methods

Cabservice() : constructor to initialize-
taxino =0, name = “”, d=0
void input() : to accept taxino, name, d

void calculate() : to calculate bill for hiring taxi as per the tariff given below:

Distance Travelled (km)	Rate/Km
Up to 1 km	25
More than 1 km and up to 5 km	30
More than 5 km and up to 10 km	35
More than 10 km and up to 20 km	40
More than 20 km	45

void display() : to display the details in the following format:

Taxino	Name	Distance (km)	Bill amount
*****	*****	*****	*****

Write the main method to create an object of a class and call all the above member methods.

```
//To calculate the billimport java.util.*;
class Cabservice
{
    String taxino,name;
    int d,amt;
    Cabservice()
    {
        taxino = " ";
        name = " ";
        d= 0; amt = 0 ;
    }
    void input()
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter taxi number: ");
        taxino = in.nextLine();
        System.out.println("Enter name of the passenger: ");
```

```

        name = in.nextLine();
        System.out.println("Enter distance travelled: ");
        d = in.nextInt();
    }
    void calculate()
    {
        if (d<=1)
            amt=25;
        if (d>1 && d<=5)
            amt=d*30;
        if (d>5 && d<=10)
            amt=d*35;
        if (d>10 && d<=20)
            amt=d*40;
        if (d>20)
            amt=d*45;
    }
    void display()
    {
        System.out.println("Taxi No"+"\\t"+"Name"+"\\t\\t"+"Distance(km)+"\\t"+"Bill
        Amount(Rs.)"); System.out.println(taxino + "\\t"+name+"\\t"+ d +"\\t\\t"+amt);
    }
    public static void main(String args[])
    {
        Cabservice ob= new Cabservice();
        ob.input();
        ob.calculate();
        ob.display();
    }

```

Output:

Enter taxi
 numberTN
 2346
 Enter name of the
 passenger: Anant
 Enter distance
 travelled:22

Taxi No	Name	Distance(km)	Bill Amount(Rs.)
TN 2346	Anant	22	990

2C. // MULTIPLE INHERITANCE

```
import java.io.*;
import java.lang.String;
class student
{
    String name;
    int regno;
    void getdata(String sname,int rno)
    {
        name=sname; regno=rno;
    }
    void putdata()
    {
        System.out.println("NAME:" +name);
        System.out.println("REGNO:" +regno);
    }
}
class mark extends student
{
    int m1,m2,m3;
    void getmarks(int mark1,int mark2,int mark3)
    {
        m1=mark1; m2=mark2; m3=mark3;
    }
    void putmarks()
    {
        System.out.println("MARK1:" +m1);
        System.out.println("MARK2:" +m2);
        System.out.println("MARK3:" +m3);
    }
}
interface s
{
    int pract_mark=60;
}
class result extends mark implements s
{
    int total;
    void display()
    {
        putdata();
        putmarks();
        System.out.println("PRACTICAL MARK:" +pract_mark);
        total=m1+m2+m3+pract_mark;
        System.out.println("TOTAL :" +total);
        if ((m1 > 40) && (m2 > 40) && (m3 > 40) && (pract_mark > 40))
            System.out.println("THE STUDENT IS PASS");
        else System.out.println("THE STUDENT IS FAIL");
    }
}
```

```

}
class list2
{
    public static void main(String args[])
    {
        result st=new result();
        st.getdata("M.SENTHILKUMAR",1001);
        st.getmarks(98,99,100);
        st.display();
    }
}

```

Output:

```

D:\jdk1.8.0_111\bin>javac list2.java
D:\jdk1.8.0_111\bin>java list2
NAME:M.SENTHILKUMAR
REGNO:1001 MARK1:98 MARK2:99 MARK3:100
PRACTICAL MARK:60
TOTAL :357
THE STUDENT IS PASS

```

2d. IMPLEMENTATION OF METHOD OVERLOADING (STATIC POLYMORPHISM/COMPILE TIME POLYMORPHISM/EARLY BINDING)

```

class ShapeCalculator
{
    // Overloaded method for calculating the area of a square
    public int area(int side)
    {
        return side * side;
    }

    // Overloaded method for calculating the area of a rectangle
    public int area(int length, int width)
    {
        return length * width;
    }

    // Overloaded method for calculating the area of a circle
    public double area(double radius)
    {
        return Math.PI * radius * radius;
    }

    // Overloaded method for calculating the area of a triangle
    public double area(double base, double height)
    {
        return 0.5 * base * height;
    }
}

```

```

    }

    public class Main
    {
        public static void main(String[] args)
        {
            ShapeCalculator calculator = new ShapeCalculator();
            // Calculate and print the area of a square
            int squareSide = 4;
            System.out.println("Area of square with side " + squareSide + ": " +
                calculator.area(squareSide));

            // Calculate and print the area of a rectangle
            int rectangleLength = 5;
            int rectangleWidth = 3;
            System.out.println("Area of rectangle with length " + rectangleLength + " and width " +
                rectangleWidth + ": " + calculator.area(rectangleLength, rectangleWidth));

            // Calculate and print the area of a circle
            double circleRadius = 2.5;
            System.out.println("Area of circle with radius " + circleRadius + ": " +
                calculator.area(circleRadius));

            // Calculate and print the area of a triangle
            double triangleBase = 6.0;
            double triangleHeight = 4.0;
            System.out.println("Area of triangle with base " + triangleBase + " and height " +
                triangleHeight + ": " + calculator.area(triangleBase, triangleHeight));
        }
    }
}

```

Output:

```

Area of square with side 4: 16
Area of rectangle with length 5 and width 3: 15
Area of circle with radius 2.5: 19.634954084936208
Area of triangle with base 6.0 and height 4.0: 12.0

```

2e. IMPLEMENTATION OF METHOD OVERRIDING (RUNTIME POLYMORPHISM/DYNAMIC BINDING)

Program creates a superclass called Figure that stores the dimensions of various two-dimensional objects. It also defines a method called area() that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and the second is Triangle. Each of these subclasses overrides area() so that it returns the area of a rectangle and triangle respectively.

```

class Figure
{
double dim1;
double dim2;

Figure(double a,double b)
{
    dim1=a;
    dim2=b;
}

double area()
{
    System.out.println("Area for Figure is undefined");
    return 0;
}
}

class Rectangle extends Figure
{
    Rectangle (double a,double b)
    {
        super(a,b);
    }
    double area()
    {
        System.out.println("Inside Area for Rectangle");
        return dim1 * dim2;
    }
}

class Triangle extends Figure
{
    Triangle (double a,double b)
    {
        super(a,b);
    }

    double area()
    {
        System.out.println("Inside Area for Triangle");
        return dim1 * dim2/2;
    }
}

class Findareas {
public static void main(String[] args) {
    Figure f=new Figure(10,10);
    Rectangle r= new Rectangle(9,5);
}
}

```



```

Triangle t=new Triangle(10,8);

Figure figref;

figref=r;
System.out.println("Area is"+figref.area());

figref=t;
System.out.println("Area is"+figref.area());

figref=f;
System.out.println("Area is"+figref.area());
}
}

```

Output:

```

Inside Area for Rectangle
Area is 45.0
Inside Area for Triangle
Area is 40.0
Area for Figure is undefined
Area is 0.0

```

2f. IMPLEMENTATION OF ABSTRACTION

Write a Java program that demonstrates abstraction through a calculator with user input. The program includes an abstract class Calculator, concrete classes for addition, subtraction, multiplication, and division, and a main class to perform these operations based on user input.

```

import java.util.Scanner;

// Abstract class Calculator
abstract class Calculator {
    double number1, number2;

    // Constructor
    public Calculator(double number1, double number2) {
        this.number1 = number1;
        this.number2 = number2;
    }

    // Abstract method for calculation
    abstract double calculate();
}

// Addition class
class Addition extends Calculator {
    public Addition(double number1, double number2) {

```

```

        super(number1, number2);
    }

    @Override
    double calculate() {
        return number1 + number2;
    }
}

// Subtraction class
class Subtraction extends Calculator {
    public Subtraction(double number1, double number2) {
        super(number1, number2);
    }

    @Override
    double calculate() {
        return number1 - number2;
    }
}

// Multiplication class
class Multiplication extends Calculator {
    public Multiplication(double number1, double number2) {
        super(number1, number2);
    }

    @Override
    double calculate() {
        return number1 * number2;
    }
}

// Division class
class Division extends Calculator {
    public Division(double number1, double number2) {
        super(number1, number2);
    }

    @Override
    double calculate() {
        if (number2 != 0) {
            return number1 / number2;
        } else {
            System.out.println("Error: Division by zero");
            return Double.NaN;
        }
    }
}

```

```

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

        // Getting user input
        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();

        System.out.print("Enter second number: ");
        double num2 = scanner.nextDouble();

        System.out.print("Enter operation (+, -, *, /): ");
        char operation = scanner.next().charAt(0);

        Calculator calculator;

        // Performing calculation based on user input
        switch (operation) {
            case '+':
                calculator = new Addition(num1, num2);
                System.out.println("Result: " + calculator.calculate());
                break;
            case '-':
                calculator = new Subtraction(num1, num2);
                System.out.println("Result: " + calculator.calculate());
                break;
            case '*':
                calculator = new Multiplication(num1, num2);
                System.out.println("Result: " + calculator.calculate());
                break;
            case '/':
                calculator = new Division(num1, num2);
                System.out.println("Result: " + calculator.calculate());
                break;
            default:
                System.out.println("Invalid operation");
                break;
        }

        scanner.close();
    }
}

```

Output:

Enter first number: 10
Enter second number: 5
Enter operation (+, -, *, /): +
Result: 15.0

Enter first number: 10
Enter second number: 5
Enter operation (+, -, *, /): -
Result: 5.0

Enter first number: 10
Enter second number: 5
Enter operation (+, -, *, /): *
Result: 50.0

Enter first number: 10
Enter second number: 5
Enter operation (+, -, *, /): /
Result: 2.0

2g. A Java program for managing airline ticketing details using inheritance. This program involves a base class Ticket, subclasses DomesticTicket and InternationalTicket, and a main class to handle user input and display the details.

```
import java.util.Scanner;
// Base class Ticket
class Ticket
{
    String passengerName;
    String flightNumber;
    double basePrice;

    // Constructor
    public Ticket(String passengerName, String flightNumber, double basePrice)
    {
        this.passengerName = passengerName;
        this.flightNumber = flightNumber;
        this.basePrice = basePrice;
    }

    // Method to display ticket details
    void displayDetails()
    {
        System.out.println("Passenger Name: " + passengerName);
        System.out.println("Flight Number: " + flightNumber);
        System.out.println("Base Price: Rs" + basePrice);
    }
}
```

```

// Subclass DomesticTicket
class DomesticTicket extends Ticket
{
    double taxRate; // Tax rate for domestic flights

    // Constructor
    public DomesticTicket(String passengerName, String flightNumber, double basePrice,
        double taxRate)
    {
        super(passengerName, flightNumber, basePrice);
        this.taxRate = taxRate;
    }

    // Overriding method to display ticket details
    @Override
    void displayDetails()
    {
        super.displayDetails();
        double totalPrice = basePrice + (basePrice * taxRate / 100);
        System.out.println("Total Price (including tax): Rs" + totalPrice);
    }
}

// Subclass InternationalTicket
class InternationalTicket extends Ticket
{
    double surcharge; // Surcharge for international flights

    // Constructor
    public InternationalTicket(String passengerName, String flightNumber, double basePrice,
        double surcharge)
    {
        super(passengerName, flightNumber, basePrice);
        this.surcharge = surcharge;
    }

    // Overriding method to display ticket details
    @Override
    void displayDetails()
    {
        super.displayDetails();
        double totalPrice = basePrice + surcharge;
        System.out.println("Total Price (including surcharge): Rs" + totalPrice);
    }
}

// Main class
public class Main
{

```

```

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

    // Getting user input for domestic ticket
    System.out.println("Enter details for Domestic Ticket:");
    System.out.print("Passenger Name: ");
    String domesticName = scanner.nextLine();
    System.out.print("Flight Number: ");
    String domesticFlight = scanner.nextLine();
    System.out.print("Base Price: ");
    double domesticBasePrice = scanner.nextDouble();
    System.out.print("Tax Rate (in %): ");
    double domesticTaxRate = scanner.nextDouble();
    scanner.nextLine(); // Consume newline

    // Creating DomesticTicket object
    DomesticTicket domesticTicket = new DomesticTicket(domesticName, domesticFlight,
    domesticBasePrice, domesticTaxRate);

    // Displaying details
    System.out.println("\nDomestic Ticket Details:");
    domesticTicket.displayDetails();

    // Getting user input for international ticket
    System.out.println("\nEnter details for International Ticket:");
    System.out.print("Passenger Name: ");
    String internationalName = scanner.nextLine();
    System.out.print("Flight Number: ");
    String internationalFlight = scanner.nextLine();
    System.out.print("Base Price: ");
    double internationalBasePrice = scanner.nextDouble();
    System.out.print("Surcharge: ");
    double internationalSurcharge = scanner.nextDouble();

    // Creating InternationalTicket object
    InternationalTicket internationalTicket = new InternationalTicket(internationalName,
    internationalFlight, internationalBasePrice, internationalSurcharge);
    // Displaying details
    System.out.println("\nInternational Ticket Details:");
    internationalTicket.displayDetails();

    scanner.close();
}
}

```

Output:

Enter details for Domestic Ticket:

Passenger Name: David

Flight Number: SwissA234

Base Price: 3456

Tax Rate (in %): 23

Passenger Name: David

Flight Number: SwissA234

Base Price: Rs3456.0

Total Price (including tax): Rs4250.88

Enter details for International Ticket:

Passenger Name: Shilpa

Flight Number: AirA320

Base Price: 8090

Surcharge: 55

International Ticket Details:

Passenger Name: Shilpa

Flight Number: AirA320

Base Price: Rs8090.0

Total Price (including surcharge): Rs8145.0