1. Create an abstract class called Bank and implement two methods AccountDetails(), LoanDetails() and one abstract method rateOfInterest() . create classes SBI , ICICI and HDFC that extends Bank abstract class and implement the rateOfInterest() . In a class CustomerAccount , create customers of SBI, ICICI and HDFC and then display account and loan details of customer along with rate of interest.
2. Create an abstract class Figure having variables dim1,dim2 of double type and an abstract method area, then make two subclasses Rectangle and Triangle which will implement the area method. Create the abstract class reference variable, assign subclass objects to it and print the corresponding area.
3. Create a program to implement the following using interface.There are three courses like DAC,DMC,DASS All the courses implements interface module which has a method placementPercentage().All the course should have member variable of course name,no of students.use parameterized constructor to set value of course name and no of students. Have a coordinator class to find out the placementPercentage of each courses and display course name,no of students placementPercentage.
4. Write a program in java which implement interface Student which has two methods Display\_Grade and Attendance for PG\_Students and UG\_Students (PG\_Students and UG\_Students are two different classes for Post Graduate and Under Graduate students respectively
5. Define an abstract class “Staff” with members name and address. Define two subclasses of this class – “FullTimeStaff” (department, salary) and “PartTimeStaff” (number of-hours, rate-per-hour). Define appropriate constructors. Create an object which could be of either FullTimeStaff or PartTimeStaff class by asking the user’s choice. Display details of all “FullTimeStaff” objects and all “PartTimeStaff” objects.
6. Define an interface “IntDefine” with methods to check whether an integer is positive, negative, even, odd, prime and operations like factorial and sum of digits. Define a class MyNumber having one private int data member. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value (Use this). Implement the above interface. Create an object in main. Perform the above operations using a menu.
7. Define an interface “StackOperations” which declares methods for a static stack. Define a class “UserStack” which contains an array and top as data members and implements the above interface. Initialize the stack using a constructor. Write a menu driven program to perform operations on a stack object

**1**

abstract class Bank

{

void account\_details()

{

String acc\_h\_name="CDAC";

String accno="3212654425";

double current\_bal=39999;

System.out.println("Account holder name="+acc\_h\_name);

System.out.println("Account number="+accno);

System.out.println("Current balance="+current\_bal);

}

void Loan\_Details()

{double current\_bal=50000;

double l\_amount=200000;

double total\_balance=(l\_amount+ current\_bal);

System.out.println("Loan amount="+ l\_amount);

System.out.println("Total balance="+total\_balance);

}

abstract void RateOfInterest();

}

class SBI extends Bank

{

void RateOfInterest()

{

double r,t=2,p=200000,si;

r=0.09;

si=(p\*r\*t);

System.out.println("Simple Interest of Bank A="+si);

}

}

class ICICI extends Bank

{

void RateOfInterest()

{

double r,t=2,p=200000,si;

r=0.08;

si=(p\*r\*t);

System.out.println("Simple Interest of Bank B="+si);

}

}

class HDFC extends Bank

{

void RateOfInterest()

{double r,t=2,p=200000,si;

r=0.1;

si=(p\*r\*t);

System.out.println("Simple Interest of Bank C="+si);

}

}

class Bank\_Details

{

public static void main(String args[])

{ SBI a=new SBI();

a.account\_details();

a.Loan\_Details();

a.RateOfInterest();

ICICI b=new ICICI();

b.RateOfInterest();

HDFC c=new HDFC();

c.RateOfInterest();

}}

2

abstract class Figure

{

void SetData()

{

double dim1=20;

double dim2=10;

System.out.println("Dimention 1="+dim1);

System.out.println("Dimention 2="+dim2);

}

abstract void Area();

}

class Rectangle extends Figure

{

void Area()

{

double dim1=20;

double dim2=10;

double area=(dim1\*dim2);

System.out.println("Area of rectangle is="+area);

}

}

class Tringle extends Figure

{

void Area()

{

double dim1=20;

double dim2=10;

double area=(0.5\*dim1\*dim2);

System.out.println("Area of Tringle is="+area);

}

}

class Figure\_area{

public static void main(String args[])

{ Rectangle a=new Rectangle();

a.SetData();

a.Area();

Tringle b=new Tringle();

b.Area();

}}

3

interface Course

{

void placementPercentage();

}

class DAC implements Course

{

String courseName;

int noOfStudents;

DAC(String courseName, int noOfStudents)

{

this.courseName=courseName;

this.noOfStudents=noOfStudents;

}

public void placementPercentage()

{

System.out.println("Course Name "+courseName+" "+"NoOfStudents"+noOfStudents);

System.out.println("DAC placement ratio is 90%");

}

}

class DMC implements Course

{

String courseName;

int noOfStudents;

DMC(String courseName, int noOfStudents)

{

this.courseName=courseName;

this.noOfStudents=noOfStudents;

}

public void placementPercentage()

{

System.out.println("Course Name"+courseName+" "+"No Of Students"+noOfStudents);

System.out.println("DMC placement ratio is 80%");

}

}

class DASS implements Course

{

String courseName;

int noOfStudents;

DASS (String courseName, int noOfStudents)

{

this.courseName=courseName;

this.noOfStudents=noOfStudents;

}

public void placementPercentage()

{

System.out.println("CourseName"+courseName+" "+"NoOfStudents"+noOfStudents);

System.out.println("DAC placement ratio is 70%");

}

public static void main(String args[])

{

DAC da=new DAC("DAC",58);

da.placementPercentage();

DMC dm=new DMC("DMC",60);

dm.placementPercentage();

DASS ds=new DASS("DASS",50);

ds.placementPercentage();

}

}

4

interface Student

{

void Display\_Grade();

void Display\_Attendance();

}

class PG\_Student implements Student

{

String name, grade;

int m1, m2, m3, attendence, total;

PG\_Student(String name, int m1, int m2, int m3, int attendence)

{

this.name = name;

this.m1 = m1;

this.m2 = m2;

this.m3 = m3;

this.attendence = attendence;

}

void Display()

{

System.out.println("Name is " + name);

System.out.println("Marks are " + m1 + " " + m2 + " " + m3);

}

public void Display\_Attendance()

{

System.out.println("The attendence is " + attendence);

}

public void Display\_Grade()

{

total = m1 + m2 + m3;

if (total > 270)

{

grade = "A";

}

else if (total < 250)

{

grade = "B";

}

else if (total < 200)

{

grade = "C";

}

else

{

grade = "D";

}

System.out.println("The Grade is " + grade);

}

}

class UG\_Student implements Student

{

String name, grade;

int m1, m2, m3, attendence, total;

UG\_Student(String name, int m1, int m2, int m3, int attendence)

{

this.name = name;

this.m1 = m1;

this.m2 = m2;

this.m3 = m3;

this.attendence = attendence;

}

void Display()

{

System.out.println("Name is " + name);

System.out.println("Marks are " + m1 + " " + m2 + " " + m3);

}

public void Display\_Attendance()

{

System.out.println("The attendence is " + attendence);

}

public void Display\_Grade()

{

total = m1 + m2 + m3;

if (total > 300)

{

grade = "S";

}

else if (total > 250)

{

grade = "A";

}

else if (total < 250)

{

grade = "B";

}

else if (total < 200)

{

grade = "C";

}

else

{

grade = "D";

}

System.out.println("The Grade is " + grade);

}

}

class StudentInformation {

public static void main(String[] args) {

PG\_Student p = new PG\_Student("Ram Gopal", 98, 88, 70, 70);

p.Display();

p.Display\_Attendance();

p.Display\_Grade();

UG\_Student u = new UG\_Student("Ravi Shankar", 80, 99, 87, 54);

u.Display();

u.Display\_Attendance();

u.Display\_Grade();

}

}

5

import java.util.Scanner;

abstract class Staff

{

String name;

String address;

void setData(String na,String address)

{

this.name=na;

this.address=address;

}

void display()

{

System.out.println(" Part time Staff Name is:"+" "+name+"Address is:"+address);

}

abstract void salary();

}

class FullTimeStaff extends Staff

{

String department;

double Salary;

FullTimeStaff(String department,double Salary)

{

this.department=department;

this.Salary=Salary;

}

void salary()

{

System.out.println("Department of worker= "+department+" Salary of Worker= "+ Salary);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*");

}

}

class PartTimeStaff extends Staff

{

double noOfHours;

int ratePerHour;

PartTimeStaff(double noOfHours,int ratePerHour)

{

this.noOfHours=noOfHours;

this.ratePerHour=ratePerHour;

}

void salary()

{

System.out.println("Number of working hour part Time Staff= "+noOfHours+" Rate

per hour is= "+ratePerHour);

double partsalary=(noOfHours\*ratePerHour);

System.out.println("Salary of Part Time Staff is: "+partsalary);

}

}

class Workers1

{

public static void main(String []args)

{

Scanner f=new Scanner(System.in);

System.out.println("Information of Full time and Part time Staff ");

System.out.println("Enter name of Fulltime staff:");

String name=f.next();

FullTimeStaff s=new FullTimeStaff("Quality",40000);

s.salary();

s.setData(“Ravi Kishan","Jaipur");

s.display();

PartTimeStaff p=new PartTimeStaff(4.5,100);

p.salary();

}

}