Name-Aditi Sudhir Igade

Div-A Roll no-221081

PRN. No-22220311

Assignment-4

**PROBLEM STATEMENT: -**

Rahul, Rohit. Rohan wants to take a loan from Banks; they approach different banks having differ ROI rates of interest. Rahul approached SBI where Rohit and Rohan approached HDFC and ICICI respectively the ROI of SBI, HDFC, ICICI is 8.4%, 7.3%, 9.7%. While each bank has a class which provides a common method getrateofinterest (). Calculate the rate of interest Rahul, Rohit and Rohan pays to banks for the loan of 5 Lakh Rs. each. Assume the required variable for calculating ROI

**OBJECTIVE: -**

To understand concepts of method overriding by writing a program for creating superclass Bank and method getrateofintrest () and overriding the same method for rate of interest calculation in subclasses.

## THEORY:-

## Method Overriding in Java

## If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.

In other words, if a subclass provides the specific implementation of the method that has been declared by one of its parent class, it is known as method overriding.

Use of Java Method Overriding

Method overriding is used to provide the specific implementation of method which is already provided by its superclass.

Method overriding is used for runtime polymorphism

**ALGORITHM: -**

1. Create Parent Class Bank.
2. Declare required variables using appropriate data type.
3. Write the method getRateofInterest() in parent class.
4. Derive the subclasses SBI, ICICI, HDFC class from parent class using extends keyword.
5. Declare the required variables and method getRateofInterest() in derived classess.
6. Display RateofInterest of all the three banks SBI, ICICI, HDFC.

**Code-**

import java.util.Scanner;

class Bank{

public double rateOfInterest;

public int time;

public double amount;

public double interest;

}

class SBI extends Bank

{

void getRateOfInterest(double n){

this.rateOfInterest=n;

}

void getTime(int t){

this.time=t;

}

void getAmount(double a){

this.amount=a;

}

void getInterest(){

interest=(double)(rateOfInterest\*time\*amount)/100f;

System.out.println(interest);

}

}

class HDFC extends Bank{

void getRateOfInterest(double n){

rateOfInterest=n;

}

void getTime(int t){

time=t;

}

void getAmount(double a){

amount=a;

}

void getInterest(){

interest=(double)(rateOfInterest\*time\*amount)/100f;

System.out.println(interest);

}

}

class ICICI extends Bank{

void getRateOfInterest(double n){

rateOfInterest=n; }

void getTime(int t){

time=t;

}

void getAmount(double a){

amount=a;

}

void getInterest(){

interest=(double)(rateOfInterest\*time\*amount)/100f;

System.out.println(interest);

}}

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter 1 for SBI bank to calculate interest");

System.out.println("Enter 2 for HDFC bank to calculate interest");

System.out.println("Enter 3 for ICICI bank to claculate interest");

int c=sc.nextInt();

double roi;

int time;

double amount;

switch(c){

case 1:

System.out.println("Enter the rate of interest of SBI bank=");

roi=sc.nextDouble();

System.out.println("Enter the time=");

time=sc.nextInt();

System.out.println("Enter the amount");

amount=sc.nextDouble();

SBI obj=new SBI();

obj.getRateOfInterest(roi);

obj.getTime(time);

obj.getAmount(amount);

obj.getInterest();

break;

case 2:

System.out.println("Enter the rate of interest of HDFC bank=");

roi=sc.nextDouble();

System.out.println("Enter the time=");

time=sc.nextInt();

System.out.println("Enter the amount");

amount=sc.nextDouble();

HDFC obj1=new HDFC();

obj1.getRateOfInterest(roi);

obj1.getTime(time);

obj1.getAmount(amount);

obj1.getInterest();

break;

case 3:

System.out.println("Enter the rate of interest of ICICI bank=");

roi=sc.nextDouble();

System.out.println("Enter the time=");

time=sc.nextInt();

System.out.println("Enter the amount");

amount=sc.nextDouble();

ICICI obj2=new ICICI();

obj2.getRateOfInterest(roi);

obj2.getTime(time);

obj2.getAmount(amount);

obj2.getInterest();

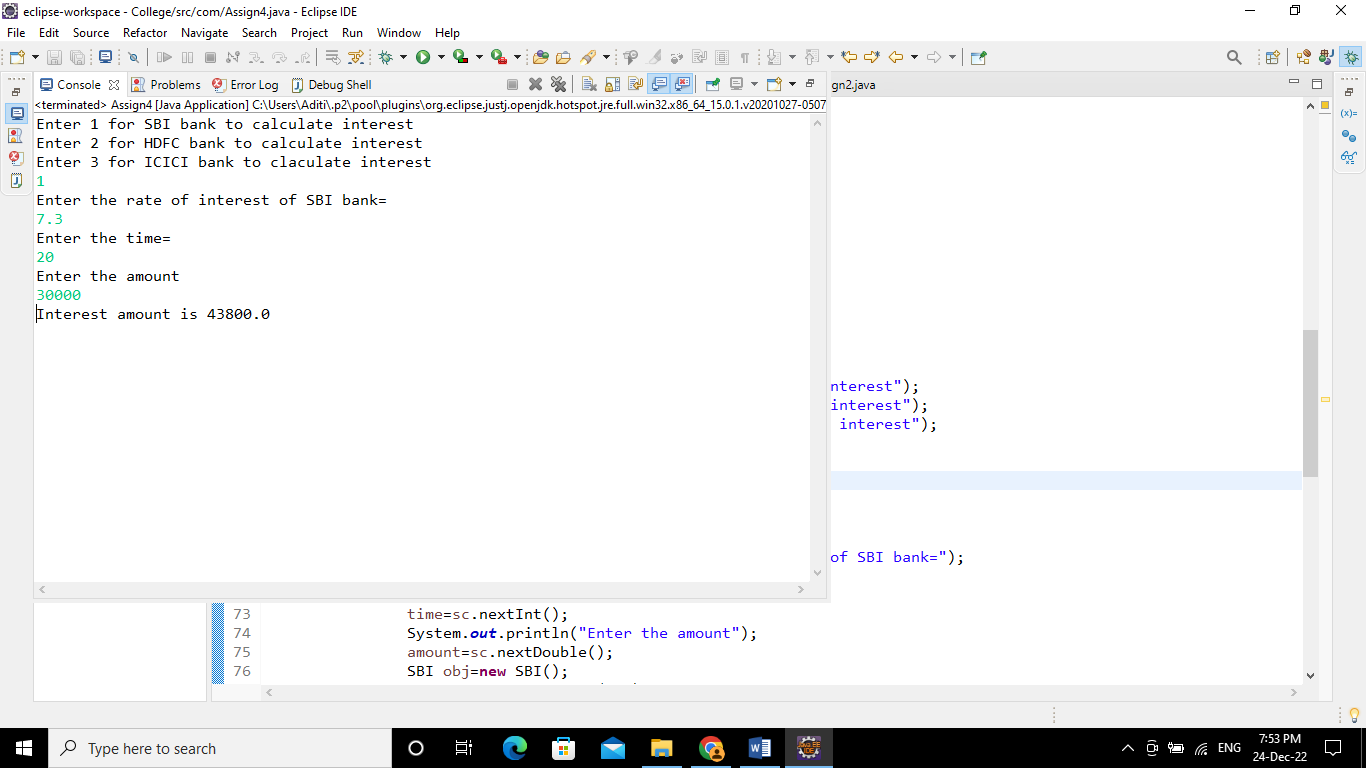
break;

}

}

}

**Output-**



**Conclusion**-Able to understand and apply the object oriented concept of method overriding.