Name	Shubhan Singh
UID no.	2022300118
Experiment No.	2

PROBLEM STATEMENT :	*A program to simulate a simple banking system in which the initial balance and rate of interest are read from the keyboard and these values are initialised using the constructor member function. The program consists of the following methods:* • *To initialise the balance amount and the rate of interest using constructor member function* • *To make deposit* • *To withdraw an amount for the balance* • *To find compound interest based on the rate of interest* • *To know the balance amount* • *To display the menu options*
THEORY:	Constructors: In Java, a constructor is a special method that is used to initialize objects when they are created. Constructors are called automatically when an object is created, and their main purpose is to initialize the instance variables of the object to appropriate values. Enhanced switch statements in java: The enhanced switch statement in Java is a more powerful and flexible version of the traditional switch statement. It was introduced in Java 12 and offers new features such as expressions as case labels and the ability to return values using the yield statement. This feature allows developers to write more concise and readable code while also improving performance. One of the main advantages of the enhanced switch statement is the ability to use expressions as case labels. In the traditional switch statement, case labels can only be constant values such as integers or strings. In contrast, the enhanced switch statement allows case labels to be expressions that are evaluated at runtime. This feature provides greater flexibility in matching values and allows developers to write more concise code. Another advantage of the enhanced switch statement is the support for multiple labels per case. This means that a single case statement can have multiple matching values, separated by commas. This feature allows developers to write

	more efficient and readable code by avoiding redundant code for similar case statements. Additionally, the enhanced switch statement supports the use of the yield statement to return a value from the switch statement. The yield statement is used within a case block to return a value, which can then be assigned to a variable or used in an expression. This feature makes the switch statement more powerful and useful for a wider range of programming scenarios.
PROGRAM:	

```
import java.util.*;

class Bank_acc{
    Scanner sc=new Scanner(System.in);
    double Balance,ROI,interest;
    int months;
    Bank_acc(double initial_Balance, double ROI){//Parametrized
constructor for initialising
        // values of initial balance and rate of interest
        Balance=initial_Balance;
        this.ROI=ROI;
    }
    void deposit(double deposited){
        Balance+=deposited;
    }//Method to add deposited amount to balance
    void Withdrawal(double withdrawn){//Method to deduct
deposited amount from balance
        if(withdrawn<=Balance){
        Balance-=withdrawn;
        }
        else{
            System.out.println("Balance insufficient for
withdrawal");
        }
    }
    void interest(){
        System.out.println("Enter time period in months for
interest calculations");
        months=sc.nextInt();
        interest=Balance*Math.pow(1+ROI/100,months)-Balance;
        System.out.println("The interest accrued in "+months+"
months is "+interest+" rupees");
    }//Method for printing compound interest(assuming monthly
interest)
    void Printbal(){
        System.out.println("The current balance in our account
is "+Balance+" rupees");
    }
    void Menu(){
        System.out.println("Select what action to perform by
entering the corresponding integer");
        System.out.print("1.Check current
balance\n2.Deposit\n3.Withdraw\n4.Calculate compound
```

```
interest\n");
        switch (sc.nextInt()) {//Enhanced switch statement,
cases defined using -> operator
            // do not require a break statement
            case 1 -> Printbal();
            case 2 -> {
                System.out.println("Enter amount to be
deposited");
                deposit(sc.nextDouble());
                System.out.println("Amount deposited");
            }
            case 3 -> {
                System.out.println("Enter amount to be
withdrawn");
                Withdrawal(sc.nextDouble());
                System.out.println("Amount withdrawn");
            }
            case 4 -> interest();
            default -> System.out.println("Invalid input");
        }
    }
}
public class Banking {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        double bal,roi;
        System.out.println("Enter initial balance in account and
monthly rate of interest(in %)");
        bal=sc.nextDouble();
        roi=sc.nextDouble();
        Bank_acc account=new Bank_acc(bal,roi);
        do{//Driver code
            account.Menu();
            System.out.println("Enter 0 to exit interface, 1 to
continue");
        }while(sc.nextInt()!=0);
    }
}
```

Link to the code(for better readability and copying): https://github.com/IAmAGoodBoy04/Java_PSOOP/blob/master/Week%202/src/Banking.java

RESULT: (on next page)

```
Enter initial balance in account and rate of interest(in %)
45000
5
Select what action to perform by entering the corresponding integer
1.Check current balance
2.Deposit
3.Withdraw
4.Calculate compound interest
2
Enter amount to be deposited
5000
Amount deposited
Enter 0 to exit interface, 1 to continue
1
Select what action to perform by entering the corresponding integer
1.Check current balance
2.Deposit
3.Withdraw
4.Calculate compound interest
1
The current balance in our account is 50000.0 rupees
Enter 0 to exit interface, 1 to continue
1
Select what action to perform by entering the corresponding integer
1.Check current balance
2.Deposit
3.Withdraw
4.Calculate compound interest
```

```
4.Calculate compound interest
3
Enter amount to be withdrawn
18000
Amount withdrawn
Enter 0 to exit interface, 1 to continue
1
Select what action to perform by entering the corresponding integer
1.Check current balance
2.Deposit
3.Withdraw
4.Calculate compound interest
1
The current balance in our account is 32000.0 rupees
Enter 0 to exit interface, 1 to continue
1
Select what action to perform by entering the corresponding integer
1.Check current balance
2.Deposit
3.Withdraw
4.Calculate compound interest
4
Enter time period in months for interest calculations
10
The interest accrued in 10 months is 20124.628056878144 rupees
Enter 0 to exit interface, 1 to continue
0

Process finished with exit code 0
```