Experiment 3

Student Name: Shivansh Ghildiyal **UID: 22BCS12928**
Branch: BE-CSE **Section/Group: EPAM 801-B**
Semester: 6th **Date of Performance: 03/02/25**
Subject Name: Project Based Learning in Java **Subject Code: 22CSH-359**

1. Aim:

To develop a Java program that calculates interest for different account types (**SB, FD, RD**) based on amount, account holder status, and duration using **OOP concepts** like **inheritance, abstraction, and method overriding**.

2. Objective:

To implement a **Java-based Interest Calculator** for different account types (**SB, FD, RD**) using **OOP principles** like **inheritance, abstraction, and method overriding**. The program should accurately compute interest based on account type, amount, and tenure while ensuring **input validation and exception handling**.

3. Implementation/Code:

```
import java.util.Scanner;
public class Exp3 {
    public static void main(String[] args) {
        System.out.println("1. Interest Calculator - SB");
        System.out.println("2. Interest Calculator - FD");
        System.out.println("3. Interest Calculator - RD");
        System.out.println("4. Exit");
        Scanner sc = new Scanner(System.in);
        int ch = sc.nextInt();
        switch (ch) {
            case 1:
            System.out.println("Enter the average amount in your account:
");
            double amount = sc.nextDouble();
            System.out.println("Enter the account type");
            String type = sc.next();
            SBAccount sb = new SBAccount(type,amount);
            System.out.println("Interest Gained: "+sb.calculateInterest());
            break;
```

```
                case 2:
                System.out.println("Enter the FD account: ");
                double amt = sc.nextDouble();
                System.out.println("Enter the number of days");
                int days = sc.nextInt();
                System.out.println("Enter the age");
                int age = sc.nextInt();
                FDAccount fd = new FDAccount(days,age,amt);
                System.out.println("Interest Gained: "+fd.calculateInterest());
                break;
                case 3:
                System.out.println("Enter the RD account: ");
                double monthlyAmount = sc.nextDouble();
                System.out.println("Enter the number of months");
                int months = sc.nextInt();
                System.out.println("Enter the age");
                int ag = sc.nextInt();
                RDAccount rd = new RDAccount(months,monthlyAmount,ag);
                System.out.println("Interest Gained: "+rd.calculateInterest());
                break;
                case 4:
                System.out.println("Exiting....");
                default:
                System.out.println("Invalid Input");}}}
                    break;
abstract class Account{
    double interestRate;
    double amount;
    abstract double calculateInterest();}
class FDAccount extends Account{
    int noOfDays;
    int ageOfACHolder;
    FDAccount(int noOfDays,int ageOfACHolder,double amount){
        this.noOfDays=noOfDays;
        this.amount=amount;
        this.ageOfACHolder=ageOfACHolder;}
    double calculateInterest(){
        if(noOfDays>=7&&noOfDays<=14){
            if(amount>10000000){
                this.interestRate=6.5;}
            else if(ageOfACHolder>=60){
                this.interestRate=5;}
```

```
            else{
                this.interestRate=4.50;}}
        else if(noOfDays>=15&&noOfDays<=29){
            if(amount>10000000){
                this.interestRate=6.75;}
            else if(ageOfACHolder>=60){
                this.interestRate=5.25;}
            else{
                this.interestRate=4.75;}}
        else if(noOfDays>=30&&noOfDays<=45){
            if(amount>10000000){
                this.interestRate=6.75;}
            else if(ageOfACHolder>=60){
                this.interestRate=6;}
            else{
                this.interestRate=5.50;}}
        else if(noOfDays>=45&&noOfDays<=60){
            if(amount>10000000){
                this.interestRate=8;}
            else if(ageOfACHolder>=60){
                this.interestRate=7.50;}
            else{
                this.interestRate=7;}}
        else if(noOfDays>=61&&noOfDays<=184){
            if(amount>10000000){
                this.interestRate=8.5;}
            else if(ageOfACHolder>=60){
                this.interestRate=8.5;}
            else{
                this.interestRate=8;}}
        else if(noOfDays>=184&&noOfDays<=365){
            if(amount>10000000){
                this.interestRate=10;}
            else if(ageOfACHolder>=60){
                this.interestRate=8.5;
            }
            else{
                this.interestRate=8;}}
        else{
            System.out.println("INVALID INPUT");
        return (amount * interestRate ) / 100;}}
    class RDAccount extends Account{
```

```
        int noOfMonths;
        int ageOfACHolder;
        double monthlyAmount;
        RDAccount(int noOfMonths,double monthlyAmount,int ageOfACHolder){
            this.noOfMonths=noOfMonths;
            this.ageOfACHolder=ageOfACHolder;
            this.monthlyAmount=monthlyAmount;
        }
        double calculateInterest(){
            if(noOfMonths==6){
                this.interestRate=ageOfACHolder>60?8:7.75;
            }
            else if(noOfMonths==9){
                this.interestRate=ageOfACHolder>60?8.25:7.75;
            }
            else if(noOfMonths==12){
                this.interestRate=ageOfACHolder>60?8.50:8;
            }
            else if(noOfMonths==15){
                this.interestRate=ageOfACHolder>60?8.75:8.25;
            }
            else if(noOfMonths==18){
                this.interestRate=ageOfACHolder>60?9:8.5;
            }
            else if(noOfMonths==21){
                this.interestRate=ageOfACHolder>60?9.25:8.75;
            }
            else{
                System.out.println("INVALID INPUT");
            }
            return monthlyAmount * noOfMonths * (interestRate / 100) *
(noOfMonths + 1) / (2 * 12);}}
    class SBAccount extends Account{
        String type;
        SBAccount(String type,double amount){
            this.type=type;
            this.amount=amount;
        }
        double calculateInterest(){
            this.interestRate=type.equalsIgnoreCase("Normal")?4:6;
            double interest = (amount * interestRate) / 100;
            return interest;}}
```

4. Output

```
Output

1. Interest Calculator - SB
2. Interest Calculator - FD
3. Interest Calculator - RD
4. Exit
Enter your choice: 2
Enter the FD amount: 50000
Enter the number of days: 180
Enter the age: 45
Interest Gained: 4000.0

=== Code Execution Successful ===
```

5. Learning Outcome

i. **Understand and implement abstraction** using abstract classes and method overriding.

ii. **Implement real-world banking scenarios** using Java OOP principles.