Experiment 3

Student Name: Satyam Kumar Rawat
Branch: CSE
Semester: 6th
Subject: Java Lab

UID: 22BCS14274
Section/Group: KRG_IOT-3B
DOP:19/02/25
Subject Code: 22CSH-359

1) Aim:

Create an application to calculate interest for FDs, RDs based on certain conditions u
inheritance

2) Objective:

To develop a Java application that calculates interest for Fixed Deposits (FDs) and Recurring Deposits (RDs) using object-oriented programming principles. The application will use inheritance to define common properties and methods for accounts while providing specific implementations for FDs and RDs based on their respective conditions.

3) Algorithm:

- Create Account class with attributes: `accountHolderName`, `principal`, `rateOfInterest`. Include methods for calculating interest (to be overridden) and displaying details.
- Create FixedDeposit subclass that calculates FD interest using: `principal * rateOfInterest * tenureInYears / 100`. Display FD details.
- Create RecurringDeposit subclass that calculates RD interest using: `monthlyDeposit * months * (months + 1) / 2) * (rateOfInterest / (12 * 100))`. Display RD details.
- In main method, create instances of `FixedDeposit` and `RecurringDeposit` and display their details.

4) Code:

```
class Account {
   String accountHolderName;
   double principal;
   double rateOfInterest;
   public Account(String accountHolderName, double principal, double rateOfInterest) {
      this.accountHolderName = accountHolderName;
      this.principal = principal;
      this.rateOfInterest = rateOfInterest;
   }
   public double calculateInterest() {
      return 0;
   }
   public void displayDetails() {
      System.out.println("Account Holder: " + accountHolderName);
      System.out.println("Principal Amount: " + principal);
      System.out.println("Rate of Interest: " + rateOfInterest + "%");
   }
}
```

```
class FixedDeposit extends Account {
    int tenureInYears;

    public FixedDeposit(String accountHolderName, double principal, double rateOfInterest, int
tenureInYears) {
        super(accountHolderName, principal, rateOfInterest);
        this.tenureInYears = tenureInYears;
    }

    @Override
    public double calculateInterest() {
        return (principal * rateOfInterest * tenureInYears) / 100;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Tenure (Years): " + tenureInYears);
        System.out.println("Interest Amount: " + calculateInterest());
    }
}

class RecurringDeposit extends Account {
    int months;
    double monthlyDeposit;
    public RecurringDeposit(String accountHolderName, double monthlyDeposit, double
rateOfInterest, int months) {
        super(accountHolderName, monthlyDeposit * months, rateOfInterest);
        this.monthlyDeposit = monthlyDeposit;
        this.months = months;
    }

    @Override
    public double calculateInterest() {
        double n = months;
        return (monthlyDeposit * n * (n + 1) / 2) * (rateOfInterest / (12 * 100));
    }

    @Override
    public void displayDetails() {
        System.out.println("Account Holder: " + accountHolderName);
        System.out.println("Total Deposited Amount: " + principal);
        System.out.println("Monthly Deposit: " + monthlyDeposit);
        System.out.println("Number of Months: " + months);
```

```
        System.out.println("Rate of Interest: " + rateOfInterest + "%");
        System.out.println("Interest Amount: " + calculateInterest());
    }
}
public class pblj {
    public static void main(String[] args) {
        FixedDeposit fd = new FixedDeposit("Baronika", 100000, 5.5, 3);
        System.out.println("Fixed Deposit Details:");
        fd.displayDetails();
        System.out.println();

        RecurringDeposit rd = new RecurringDeposit("Baronika", 5000, 6.5, 12);
        System.out.println("Recurring Deposit Details:");
        rd.displayDetails();
    }
}
```

5) Output:

```
PS C:\Users\hp.pc\Desktop\programming languages\java\javaClass> cd "c:\
 ; if ($?) { javac Pro.java } ; if ($?) { java Pro }
Fixed Deposit Details:
Account Holder: Anushka
Principal Amount: 100000.0
Rate of Interest: 5.5%
Tenure (Years): 3
Interest Amount: 16500.0

Recurring Deposit Details:
Account Holder: Anushka
Total Deposited Amount: 60000.0
Total Deposited Amount: 60000.0
Monthly Deposit: 5000.0
Total Deposited Amount: 60000.0
Monthly Deposit: 5000.0
Total Deposited Amount: 60000.0
Total Deposited Amount: 60000.0
Monthly Deposit: 5000.0
Number of Months: 12
Rate of Interest: 6.5%
Interest Amount: 2112.5
PS C:\Users\hp.pc\Desktop\programming languages\java\javaClass>
```

6) Learning Outcomes:

1) Inheritance: Use of base and derived classes for shared attributes and methods.
2) Method Overriding: Custom implementation of methods in subclasses.
3) Constructor: Initializing object attributes using constructors.
4) Encapsulation: Storing and manipulating data within objects.
5) Polymorphism: Different behavior of calculateInterest() based on object type.
6) Interest Calculation: Implementing FD and RD interest formulas.
7) Class Interaction: Creating objects and calling methods to display details.