

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 3

Name: Kartiakay Awasthi

Branch: CSE

Semester: 6th

Subject: Java

UID: 22BCS13902

Section: IOT-631 /A

DOP: 13/02/2025

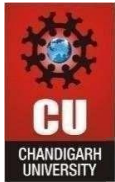
Subject Code: 22CSH-359

1. Aim: Calculate interest based on the type of the account and the status of the account holder. The rates of interest changes according to the amount (greater than or less than 1 crore), age of account holder (General or Senior citizen) and number of days if the type of account is FD or RD.

2. Objective: The objective is to calculate interest for an account based on its type (e.g., FD, RD, Savings) and the account holder's status (General or Senior Citizen). The interest rate varies depending on factors such as the account balance (greater than or less than 1 crore), the account holder's age, and the duration (number of days) for fixed or recurring deposits. The system dynamically determines the applicable interest rate and computes the interest accordingly.

3. Algorithm:

1. Input: Account type, account holder status (General/Senior Citizen), account balance, and duration (if FD/RD).
2. Check Account Type:
3. If Savings Account:
4. Use predefined interest rates based on balance (>1 crore or <1 crore) and status.
5. If FD/RD:
6. Use interest rates based on duration, balance (>1 crore or <1 crore), and status.
7. Determine Interest Rate:
8. Fetch the applicable rate from the rate table based on balance, status, and duration (if applicable).
9. Calculate Interest:
10. For Savings Account:
11. $\text{Interest} = \text{Balance} \times \text{Interest Rate} \times \text{Duration (in years)}$
12. $\text{Interest} = \text{Balance} \times \text{Interest Rate} \times 1 \text{Duration (in years)}$
13. For FD/RD:
14. $\text{Interest} = \text{Principal} \times \text{Interest Rate} \times \text{Duration (in days)} / 365$
15. $\text{Interest} = \text{Principal} \times \text{Interest Rate} \times 365 \text{Duration (in days)}$
16. Output: Return the calculated interest.

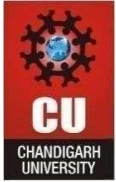


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Code:

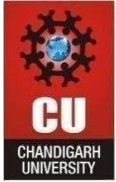
```
package Package2;
import java.util.*;
abstract class Account
{
    double interestRate;
    double amount;
    abstract double calculateInterest();
}
class SBAccount extends Account
{
    public SBAccount(double amount)
    {
        this.amount = amount;
        this.interestRate = 0.04;
    }
    public double calculateInterest()
    {
        return this.amount * this.interestRate;
    }
}
class FDAccount extends Account
{
    int noOfDays;
    int ageOfACHolder;
    public FDAccount(double amount, int noOfDays, int ageOfACHolder)
    {
        this.amount = amount;
        this.noOfDays = noOfDays;
        this.ageOfACHolder = ageOfACHolder;
    }
    public double calculateInterest()
    {
        if (amount < 0 || noOfDays < 0 || ageOfACHolder < 0)
        {
            throw new IllegalArgumentException("Invalid values entered.");
        }
        double interestRate = 0;
        if (amount < 1_00_00_000)
        {
            if (noOfDays >= 7 && noOfDays <= 14) interestRate = ageOfACHolder >= 60 ? 5.00 : 4.50;
            else if (noOfDays >= 15 && noOfDays <= 29) interestRate = ageOfACHolder >= 60 ? 5.25 :
4.75;
            else if (noOfDays >= 30 && noOfDays <= 45) interestRate = ageOfACHolder >= 60 ? 6.00 :
5.50;
            else if (noOfDays >= 45 && noOfDays <= 60) interestRate = ageOfACHolder >= 60 ? 7.50 :
7.00;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

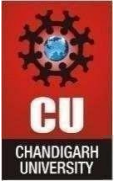
```
        else if (noOfDays >= 61 && noOfDays <= 184) interestRate = ageOfACHolder >= 60 ? 8.00 :
7.50;
        else if (noOfDays >= 185 && noOfDays <= 365) interestRate = ageOfACHolder >= 60 ? 8.50
: 8.00;
    }
    else
    {
        if (noOfDays >= 7 && noOfDays <= 14) interestRate = 6.50;
        else if (noOfDays >= 15 && noOfDays <= 29) interestRate = 6.75;
        else if (noOfDays >= 30 && noOfDays <= 45) interestRate = 6.75;
        else if (noOfDays >= 45 && noOfDays <= 60) interestRate = 8.00;
        else if (noOfDays >= 61 && noOfDays <= 184) interestRate = 8.50;
        else if (noOfDays >= 185 && noOfDays <= 365) interestRate = 10.00;
    }
    return (amount * interestRate) / 100;
}
}
class RDAccount extends Account {
    int noOfMonths;
    double monthlyAmount;
    public RDAccount(double monthlyAmount, int noOfMonths) {
        this.monthlyAmount = monthlyAmount;
        this.noOfMonths = noOfMonths;
    }
    @Override
    public double calculateInterest() {
        if (monthlyAmount < 0 || noOfMonths < 0) {
            throw new IllegalArgumentException("Invalid values entered.");
        }
        double rate = 0;
        if (noOfMonths == 6) rate = 7.50;
        else if (noOfMonths == 9) rate = 7.75;
        else if (noOfMonths == 12) rate = 8.00;
        else if (noOfMonths == 15) rate = 8.25;
        else if (noOfMonths == 18) rate = 8.50;
        else if (noOfMonths == 21) rate = 8.75;
        return (monthlyAmount * noOfMonths * rate) / 100;
    }
}
class InvalidInputException extends Exception {
    public InvalidInputException(String message) {
        super(message);
    }
}
public class Assignment3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("Select the option:");
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
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");
int choice = scanner.nextInt();
try {
    if (choice == 1) {
        System.out.println("Enter the Average amount in your account:");
        double amount = scanner.nextDouble();
        if (amount < 0) throw new InvalidInputException("Amount cannot be negative.");
        SBAccount sbAccount = new SBAccount(amount);
        System.out.println("Interest gained: Rs. " + sbAccount.calculateInterest());
    } else if (choice == 2) {
        System.out.println("Enter the FD amount:");
        double amount = scanner.nextDouble();
        System.out.println("Enter the number of days:");
        int days = scanner.nextInt();
        System.out.println("Enter your age:");
        int age = scanner.nextInt();
        if (amount < 0 || days < 0 || age < 0) throw new InvalidInputException("Invalid input
values.");
        FDAccount fdAccount = new FDAccount(amount, days, age);
        System.out.println("Interest gained: Rs. " + fdAccount.calculateInterest());
    } else if (choice == 3) {
        System.out.println("Enter the monthly RD amount:");
        double amount = scanner.nextDouble();
        System.out.println("Enter the number of months:");
        int months = scanner.nextInt();
        if (amount < 0 || months < 0) throw new InvalidInputException("Invalid input values.");
        RDAccount rdAccount = new RDAccount(amount, months);
        System.out.println("Interest gained: Rs. " + rdAccount.calculateInterest());
    } else if (choice == 4) {
        break;
    } else {
        System.out.println("Invalid choice. Please try again.");
    }
} catch (InvalidInputException e) {
    System.out.println(e.getMessage());
}
}
scanner.close();
}
```

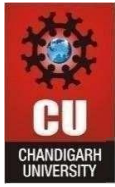


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output:

```
Assignment3 [Java Application] C:\Users\SHIVAM\.p2\pool\plugins\org.eclipse.justj.open  
Select the option:  
1. Interest Calculator -SB  
2. Interest Calculator -FD  
3. Interest Calculator -RD  
4. Exit  
1  
Enter the Average amount in your account:  
10000  
Interest gained: Rs. 400.0  
Select the option:  
1. Interest Calculator -SB  
2. Interest Calculator -FD  
3. Interest Calculator -RD  
4. Exit  
2  
Enter the FD amount:  
2000  
Enter the number of days:  
30  
Enter your age:  
21  
Interest gained: Rs. 110.0  
Select the option:  
1. Interest Calculator -SB  
2. Interest Calculator -FD  
3. Interest Calculator -RD  
4. Exit
```



COMPUTER SCIENCE & ENGINEERING

DEPARTMENT OF

Discover. Learn. Empower.

Learning Outcomes:

- **Object-Oriented Design:** Learn to create and use classes for real-world entities.
- **Core Programming Skills:** Practice loops, conditionals, and methods for inventory operations.
- **Data Structure Usage:** Use `ArrayList` to manage dynamic data effectively.
- **User-Friendly Systems:** Design intuitive interfaces and handle errors smoothly.