

Title of the Experiment: Implementation of parameterized methods and constructors**Experiment No.3****Date:** 24/10/2020**Problem Statement:**

A certain small bank intends to automate few of its banking operations for its customers.

Design a class by name mybankAccount to store the customer data having following details:

1.accountNumber 2. acctType 3. Name 4. Address 5. accountBalance

The class must have both default and parameterized constructors. Write appropriate method to compute interest accrued on accountBalance based on accountType and time in years. Assume 5% for S/B account 6.5% for RD account and 7.65 for FD account. Further, add two methods withdrawAmount/depositAmount with amount as input to withdraw and deposit respectively.

The withdrawAmount method must report insufficient balance if accountBalance falls below Rs. 1000.

TASK 1: Build the class with appropriate member variables, constructors and methods.

TASK 2: Instantiate three objects of above type and perform different operations on the same.

TASK 3: Write a function to display all the three customer details in a tabular form with appropriate column headings.

Objectives of the Experiment:

1. Learn declaration and definition of constructors and methods in Java
2. Learn how to declare objects of a class and access methods
3. Learn how constructors are executed and its different types
4. Understand the use of constructors and methods in a real-life application
5. Learn how to pass parameters to a method and constructors
6. Learn to Display the result in a readable/proper format

Program Source Code:

```
import java.util.*;
class myBankAccount{
    int accNo;
    String name,address,acctType;
    double balance;
    static int count=0;
```

```

myBankAccount(){
    Scanner in= new Scanner(System.in);
    accNo=++count;
    System.out.println("Enter the type of account(SB | FD | RD): ");
    accType=in.nextLine();
    System.out.println("Enter customer name: ");
    name=in.nextLine();
    System.out.println("Enter customer address: ");
    address=in.nextLine();
    System.out.println("Enter Initial Amount: ");
    balance=in.nextDouble();
}

myBankAccount(String type, String name, String address, double balance){
    accNo=++count;
    accType=type;
    this.name=name;
    this.address=address;
    this.balance=balance;
}

void computeInterest(){
    double interest=0;
    switch(accType){
        case "SB" :{
            interest=balance*0.05;
            break;
        }
        case "RD" :{
            interest=balance*0.065;
            break;
        }
        case "FD" :{
            interest=balance*0.0765;
            break;
        }
    }
    System.out.println("Interest Earned: " + interest);
}

void withdrawAmount(double amt){
    if(accType.equals("SB")){
        if(balance-amt<1000)
            System.out.println("Denied...Insufficient Balance");
        else
            balance=balance-amt;
    }
}

```

```

        else
            System.out.println("Withdrawal not permitted on these types of account");
    }
    void depositAmount(double amt){
        balance=balance+amt;
    }
    void dispDetails(){
        System.out.println(String.format("%-10s|%-10d|%5s|%-6.2f",name,accNo,accType,balance));
    }
}

public class Bank{
    public static void main(String[] args){
        myBankAccount a1= new myBankAccount("SB","Venkatesh","Bhagyanagar Belagavi",348732.00);
        myBankAccount a2= new myBankAccount("FD","Rohit","Samrudhi ColonyBelagavi",254321.00);
        myBankAccount a3= new myBankAccount("RD","Ajay","Bhagyanagar Belagavi",2354234.00);

        System.out.println(String.format("%-10s|%-10s|%5s|%-6s","Name","AccNumber","Type","Balance"));
        a1.dispDetails();
        a2.dispDetails();
        a3.dispDetails();

        Scanner in= new Scanner(System.in);
        myBankAccount a4= new myBankAccount();
        a4.computeInterest();
        double amnt;
        System.out.println("Enter the amount to be deposited: ");
        amnt=in.nextDouble();
        a4.depositAmount(amnt);

        System.out.println("Enter the amount to be withdrawn: ");
        amnt=in.nextDouble();
        a4.withdrawAmount(amnt);

        System.out.println(String.format("%-10s|%-10s|%5s|%-6s","Name","AccNumber","Type","Balance"));
        a4.dispDetails();
    }
}

```

OUTPUT:

```
Name      |AccNumber | Type|Balance
Venkatesh |1         | SB|348732.00
Rohit     |2         | FD|254321.00
Ajay      |3         | RD|2354234.00
Enter the type of account(SB | FD | RD):
SB
Enter customer name:
Sanket
Enter customer address:
Mahantesh Nagar
Enter Initial Amount:
10000
Interest Earned: 500.0
Enter the amount to be deposited:
400
Enter the amount to be withdrawn:
100
Name      |AccNumber | Type|Balance
Sanket    |4         | SB|10300.00
PS D:\Programming\JAVA\Java\jdk\bin> |
```

```
Name      |AccNumber | Type|Balance
Venkatesh |1         | SB|348732.00
Rohit     |2         | FD|254321.00
Ajay      |3         | RD|2354234.00
Enter the type of account(SB | FD | RD):
FD
Enter customer name:
Ayan
Enter customer address:
Shri Nagar
Enter Initial Amount:
200000
Interest Earned: 15300.0
Enter the amount to be deposited:
1000
Enter the amount to be withdrawn:
500
Withdrawal not permitted on these types of account
Name      |AccNumber | Type|Balance
Ayan      |4         | FD|201000.00
PS D:\Programming\JAVA\Java\jdk\bin> |
```

```
PS D:\Programming\JAVA\Java\jdk\bin> java app
Name      |AccNumber | Type|Balance
Venkatesh |1         | SB|348732.00
Rohit     |2         | FD|254321.00
Ajay      |3         | RD|2354234.00
Enter the type of account(SB | FD | RD):
RD
Enter customer name:
Sejal
Enter customer address:
Anjaneya Nagar
Enter Initial Amount:
900000
Interest Earned: 58500.0
Enter the amount to be deposited:
50000
Enter the amount to be withdrawn:
10
Withdrawal not permitted on these types of account
Name      |AccNumber | Type|Balance
Sejal     |4         | RD|950000.00
|
```

Outcomes of the Experiment: At the end of the laboratory sessions the students should be able to

1. Demonstrate the use of constructors and methods in solving real-life problems.
2. Identify when to use methods and when to use constructors
3. Identify appropriate method or constructor to be used for a particular scenario
4. Learn how to return values from a method
5. Differentiate between default and parameterized constructors
6. Identify how to make a call for the method

Conclusions: From the given problem statement, we could identify the necessary variables and use the appropriate constructors and methods and the necessary program logic. We understood how the banking system works and the way in which transactions are handled. The program was written in Visual Studio Code by creating a project. We understood the usage of the IDE in typing the code, debugging, running the program and observing the output. We also understood the use of the built-in class System and its method println to display the result. The program was executed for two-three sets of input and results obtained were verified to be correct and recorded.

PRACTICE PROBLEM

Define a class to represent a rectangle in which constructors and parameterized methods are used. It also has a method to compute area of rectangle.

- i. First make a class rectangle in which we declare the parameterized constructor.
- ii. Then demonstrate the use of parameterized method.
- iii. Use a method to compute area of rectangle.
- iv. Create a class to demonstrate the call of the methods in previously created class rectangle holding constructors, parameterized methods and method to compute area of rectangle.

Program Source Code:

```
import java.util.Scanner;

class Rectangle{
    int length,breadth;
    Rectangle(int l,int b){
        length=l;
        breadth=b;
    }
    Rectangle(){
        //initialize an object
    }
    void changeDim(int length,int breadth) {
        this.length=length;
        this.breadth=breadth;
    }
    int area() {
        return length*breadth;
    }
}

public class tw3 {
```

```

public static void main(String[] args) {
    Rectangle r1=new Rectangle(20,40);
    System.out.println("Area of rectangle for object r1 :"+r1.area());
    Scanner s=new Scanner(System.in);
    Rectangle r2=new Rectangle();
    System.out.println("Enter the length and breadth for rectangle: ");
    int length=s.nextInt();
    int breadth=s.nextInt();
    r2.changeDim(length, breadth);
    System.out.println("Area of rectangle for object r2 :"+r2.area());
}
}

```

OUTPUT:

```

PS D:\Programming\JAVA\Java\jdk\bin> java app
Area of rectangle for object r1 :800
Enter the length and breadth for rectangle:
90
50
Area of rectangle for object r2 :4500

```

```

Area of rectangle for object r1 :800
Enter the length and breadth for rectangle:
35 45
Area of rectangle for object r2 :1575
PS D:\Programming\JAVA\Java\jdk\bin>

```