# Experiment 4

# Class To Represent Bank Account

**Date Of Submission:** 17-09-2020

**Aim:** Design a class to represent a bank account. Include the following members.

Data Members:

- Name of the depositor
- Account Number
- Type of Account
- Balance amount in the account

Methods

- To deposit an amount
- To withdraw an amount after checking balance
- To display the name and balance

Incorporate default and parameterized constructor to provide initial values

**Concept Used:** Class, Constructor Overloading, Polymorphism

**Algorithms :**

Algorithm : depositAmount(x):

Step 1: Start
Step 2: balance += x
Step 3: Stop

Algorithm :withdrawAmount(x):

Step 1: Start
Step 2: balance -= x
Step 3: Stop

Algorithm : display()

Step 1: Start
Step 2: print name
Step 3: print accountNumber
Step 4: print account type
Step 5: print balance
Step 6: Stop

Algorithm Bank() //Default Constructor

Step 1: start
Step 2: name = “”
Step 3: accountNumber =0
Step 4: accountType = “”
Step 5: balance=0
Step 6: Stop

Algorithm Bank(n,num,type,bal)

Step 1: Start
Step 2: name = n
Step 3: accountNumber = num
Step 4: accountType = type
Step 5: balance=bal
Step 6: Stop

**Result:** The program was successfully compiled and the required output was obtained

**Program:**

```
/**
  BankAccount.java
  Created By: Rohit Karunakaran

 **/

class BankAccount{
    String name;
    long accNumber;
    String accType;
    float balance;

    public void withdraw(float amount){
        //You cannot withdraw negative values
        if(amount>=0){

            if(balance-amount > 0){
                balance-=amount;
                System.out.println("Sucessfully withdrawn "+ amount+" from the
account");
            }
            else{
                System.out.println("Error! Insufficient balance in Account");
            }
            System.out.println("Remaining balance in Account = " + balance+"\n");
        }

        else{
            System.out.println("Please enter a valid amount\n");
        }
    }

    public void deposit(float amount){
        //You can't deposit negative values
        if(amount>0){
            balance+=amount;
            System.out.println("Successfully deposited "+amount+" to the account");
            System.out.println("Remaining balance is "+ balance+"\n");
        }
        else{
            System.out.println("Error!!!! Enter a valid value\n");
        }
    }
```

```
public void display(){
    System.out.println("\n=================================================");
    System.out.println("Account Number: " + accNumber);
    System.out.println("Name of the Account Holder: "+ name);
    System.out.println("Account Type: "+ accType);
    System.out.println("Balance: "+ balance);
    System.out.println("=================================================\n");
}
//Default constructor
BankAccount(){
    name = " ";
    accNumber = 0l;
    accType = " ";
    balance = 0.0f;
}

//Paramaterised constructor
BankAccount(String n,float b, long a, String t){
    name = n;
    balance = b;
    accType = t;
    accNumber = a;
}

public static void main(String[] args){
    long a;
    String name;
    String type;
    float cash;

    BankAccount benk = new BankAccount();
    BankAccount benk1 = new BankAccount("Babu",8034.123f,1029343458l,"Savings");

    benk.display();
    benk1.display();

    a = 11066049032l;
    name = "Dhamodharan";
    cash = 2340.0f;
    type = "Checkings";

    benk.name = name;
    benk.accNumber=a;
    benk.balance=cash;
    benk.accType=type;

    benk.withdraw(9000.0f);
    benk1.withdraw(800f);
    benk.deposit(-2000f);

    benk.display();
    benk1.display();
}
```

```
}
```

**Sample Output:**

```
==================================================
Account Number: 0
Name of the Account Holder:
Account Type:
Balance: 0.0
==================================================


==================================================
Account Number: 1029343458
Name of the Account Holder: Babu
Account Type: Savings
Balance: 8034.123
==================================================

Error! Insufficient balance in Account
Remaining balance in Account = 2340.0

Sucessfully withdrawn 800.0 from the account
Remaining balance in Account = 7234.123

Error!!!! Enter a valid value


==================================================
Account Number: 11066049032
Name of the Account Holder: Dhamodharan
Account Type: Checkings
Balance: 2340.0
==================================================


==================================================
Account Number: 1029343458
Name of the Account Holder: Babu
Account Type: Savings
Balance: 7234.123
==================================================
```

# Experiment 5

# Employee Class

**Date Of Submission: 17-09-2020**

**Aim:** Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same.

**Concepts Used:** Class, Inheritance, Method Overloading, this keyword

**Algorithms:**

Class Employee
1. Declare fields name, age, phone, address, salary
2. Define printSalary and printDetails methods

Class Officer inherits Class Employee
1. Declare additional field specialization
2. Define constructors for with and without specialization

Class Manager inherits Class Employee
1. Declare additional field department
2. Define constructors for with and without department

**Result:** The program is successfully compiled and the required output is obtained

**Program:**

```
/*  Employee.java
 *
 *  Done By: Rohit Karunakaran
 *
 *  Details On Classes:
 *
 *  Employee:
 *      Attributes:
 *          Name -- String
 *          Age -- int
 *          PhoneNumber -- long
 *          Address -- String
 *          Salary -- float
 *      Methods:
 *          printSalary() -- prints the salary of the employee
 *
 *  Officer - inherited from Employee:
 *      Attributes:
 *          specialization -- String
 *
 *  Manager - inherited from Employee:
```

```
 *      Attributes:
 *          department -- String
 *
 * Task: assign name, age, phoneNo, addr and salary to an office and a manager
 *      by making an object of both and print the same
 * */


class Employee{
    String name;
    int age;
    long phoneNo;
    String addr;
    float salary;

    void displayEmployee(){
        System.out.println("Name: "+this.name);
        System.out.println("Address: "+this.addr);
        System.out.println("Phone Number: "+this.phoneNo);
        System.out.println("Salary : "+this.salary);
    }
       //Function to display the salary of the Employee
    void printSalary(){
        System.out.println("Salary of "+this.name+" is "+ this.salary);
    }

    Employee(String name, int age, long phoneNo, String addr,float salary){
        this.name = name;
        this.age = age;
        this.phoneNo = phoneNo;
        this.addr = addr;
        this.salary = salary;
    }

    //Default Constructor
       Employee(){
        this(" ",0,0l," ",0.0f);
    }

    public static void main(String [] args){
        Manager m1 = new Manager("Radhakrishnan",59,9823148320l,"123, Boulevard of Broken
Dreams",4529.19f,"Research and Development");
        Officer o1 = new Officer("Anto Davis",48, 9847120926l,"Green House Villa,
Pulmaidhanam", 3892.81f,"Corporate Security");
        m1.printSalary();
        o1.printSalary();

        m1.displayEmployee();
        o1.displayEmployee();
    }

}

class Officer extends Employee{
    String spec;

    void displayEmployee(){
        System.out.println("========================================");
        super.displayEmployee();
```

```
        System.out.println("Specialization in "+ this.spec);
        System.out.println("========================================");
        System.out.println("\n");
    }

    Officer(String name, int age, long phoneNo, String addr,float salary,String spec){
        super(name,age,phoneNo,addr,salary);
        this.spec = spec;
    }

    Officer(){
        this(" ",0,0l," ",0.0f," ");
    }
}

class Manager extends Employee{
    String dep;

    void displayEmployee(){
        System.out.println("========================================");
        super.displayEmployee();
        System.out.println("Manager of "+ this.dep);
        System.out.println("========================================");
        System.out.println("\n");
    }

    Manager(String name, int age, long phoneNo, String addr,float salary,String dep){
        super(name,age,phoneNo,addr,salary);
        this.dep= dep;
    }

    Manager(){
        this(" ",0,0l," ",0.0f," ");
    }

}
```

**Sample Output:**

```
Salary of Radhakrishnan is 4529.19
Salary of Anto Davis is 3892.81
========================================
Name: Radhakrishnan
Address: 123, Boulevard of Broken Dreams
Phone Number: 9823148320
Salary : 4529.19
Manager of Research and Development
========================================


========================================
Name: Anto Davis
Address: Green House Villa, Pulmaidhanam
Phone Number: 9847120926
Salary : 3892.81
Specialization in Corporate Security
========================================
```