Experiment No. 1

Title: Program based on fundamental object oriented methodology concepts.

Batch:SY_IT(B3) **Roll No.: 16010423076** **Experiment No.:1**

Aim: Write a program that accepts customer account information
1) Customer Name
2) Account Number
and provides below menu driven operations on customer account
1) Deposit
2) Withdraw
3) Display Balance
(Create appropriate classes and methods to perform above operations.)

Resources needed: Java

Theory:

Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language.

Object in Java

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible). The example of an intangible object is the banking system.

An object has three characteristics:

- **State:** represents the data (value) of an object.
- **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw, etc.
- **Identity:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

For Example, Pen is an object. Its name is Reynolds; color is white, known as its state. It is used to write, so writing is its behavior.

An object is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

Object Definitions:

- An object is a real-world entity.
- An object is *a* runtime entity.
- The object is an entity which has state and behavior.
- The object is an instance of a class.

Class in Java

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.
A class in Java can contain:

- **Fields**
- **Methods**
- **Constructors**
- **Blocks**
- **Nested class and interface**

Syntax to declare a class:

```
class <class_name>{
   field;
   method;
}
```

Instance variable in Java

A variable which is created inside the class but outside the method is known as an instance variable. Instance variable doesn't get memory at compile time. It gets memory at runtime when an object or instance is created. That is why it is known as an instance variable.

- Method in Java

In Java, a method is like a function which is used to expose the behavior of an object.
Advantage of Method: Code Reusability, Code Optimization

- new keyword in Java

The new keyword is used to allocate memory at runtime. All objects get memory in Heap memory area.

- Object and Class Example:

```
class Student{
 int rollno;
 String name;
 void insertRecord(int r, String n){
  rollno=r;
  name=n;
 }
 void displayInformation(){System.out.println(rollno+" "+name);}
}
class TestStudent4{
 public static void main(String args[]){
  Student s1=new Student();
  Student s2=new Student();
  s1.insertRecord(111,"Karan");
  s2.insertRecord(222,"Aryan");
  s1.displayInformation();
  s2.displayInformation();
 }
}
```

Results: (Program Code with output)

Code/Input :

```
import java.util.Scanner;
public class BankMenu {
    String name;
    int accNum;
    int bal;

    void insertRecord(String n, int inpacc)
    {
    name = n;
    accNum = inpacc;
    bal = 0;
    }
    void deposit(int amt)
    {
    bal = bal+amt;
    System.out.println("Deposited : " +amt);
    }
    void withdraw(int amt)
    {
    bal = bal-amt;
    System.out.println("Withdrawn : " +amt);
    }
    void display()
    {
    System.out.println("Account balance : " +bal);
    }

    public static void main(String argz[])
    {
        Scanner sc = new Scanner(System.in);
        BankMenu cus = new BankMenu();
        System.out.print("Enter customer name : ");
        String name = sc.nextLine();
        System.out.print("Enter account number : ");
        int inpacc = sc.nextInt();
        cus.insertRecord(name, inpacc);

        boolean whileloop = true;
        while(whileloop)
        {
            System.out.println("\n1 - deposit");
            System.out.println("2 - withdraw");
            System.out.println("3 - display Balance");
            System.out.println("4 - exit");
            System.out.print("1/2/3/4 : ");
            int choice = sc.nextInt();
            if(choice == 1)
            {
            System.out.print("Enter deposit amount : ");
            int amount = sc.nextInt();
            cus.deposit(amount);
            }
            else if(choice == 2)
            {
            System.out.print("Enter amount to withdraw : ");
            int amount = sc.nextInt();
            cus.withdraw(amount);
            }
            else if(choice == 3)
            {
            cus.display();
            }
            else if(choice == 4)
            {
            break;
            }
        }
        sc.close();
    }
}
```

Output :

```
Enter customer name : Aksh
Enter account number : 3939

1 - deposit
2 - withdraw
3 - display Balance
4 - exit
1/2/3/4 : 1
enter deposit amount : 500
Deposited : 500

1 - deposit
2 - withdraw
3 - display Balance
4 - exit
```

```
1/2/3/4 : 3
Account balance : 500

1 - deposit
2 - withdraw
3 - display Balance
4 - exit
1/2/3/4 : 2
Enter amount to withdraw : 49
Withdrawn : 49

1 - deposit
2 - withdraw
3 - display Balance
4 - exit
1/2/3/4 : 3
Account balance : 451

1 - deposit
2 - withdraw
3 - display Balance
4 - exit
1/2/3/4 : 4

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

Questions:

1. Explain the constructors in java with appropriate example?

   The constructor is a class method in Java used to initialize objects. A constructor is invoked at the time of the creation of an object of the class. It sets the initial values for attributes that an object holds, and it is defined with the same name as that of the class.

   Types of Constructors

1. Default Constructor
2. Parameterized Constructor
3. Copy Constructor
4. Overloaded Constructor

Example : (The below example covers all 4 types)

```
class Constructor
{
        int pen,pencil;
        public Constructor()
        {
                pen = 10;
                pencil = 5;
        }
        public Constructor(int p, int pc)
        {
                pen = p;
                pencil = pc;
        }

        public void display()
        {
                System.out.println("The price of Pen is : Rs." +pen);
                System.out.println("The price of Pencil is : Rs." +pencil);
        }

        public static void main(String args[])
        {
                Constructor obj1 = new Constructor();
                obj1.display();
                Constructor obj2 = new Constructor();
                obj2.display();
                obj1 = obj2;
                obj1.display();
        }
}
```

Outcomes:

CO1: Apply fundamental Object Oriented Methodology concepts using java programming

Conclusion: (Conclusion to be based on the outcomes achieved)

From this article I learned how to create and use classes and objects in Java. I understood how to handle user input with the Scanner class, implement methods for different functionalities and use loops and conditional statements for program flow. This helped me grasp basic object-oriented programming.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References Books

1. Herbert Schildt; JAVA The Complete Reference; Seventh Edition, Tata McGraw-Hill Publishing Company Limited 2007.
2. Java 7 Programming - Black Book : Kogent Learning Solutions Inc.
3. Sachin Malhotra, Saurabh Chaudhary “Programming in Java”, Oxford University Press, 2010
4. Jaime Nino, Frederick A. Hosch, ‘An introduction to Programming and Object Oriented Design using Java’, Wiley Student Edition.