K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)
Department of Computer Engineering

Batch:	**D3**	**Roll No.:**	**16010123294**

Experiment / assignment / tutorial No. 05

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

Signature of the Staff In-charge with date

TITLE : An Array of Objects

AIM: Write a program which accepts information about n no of customers from user.

Create an array of objects to store account_id, name, and balance.

Your program should provide following functionalities

1. To add account

2. To delete any account detail

3. To display account details.

Expected OUTCOME of Experiment:

CO1: Apply the features of object oriented programming languages. (C++ and Java)
CO2:Explore arrays, vectors, classes and objects in C++ and Java

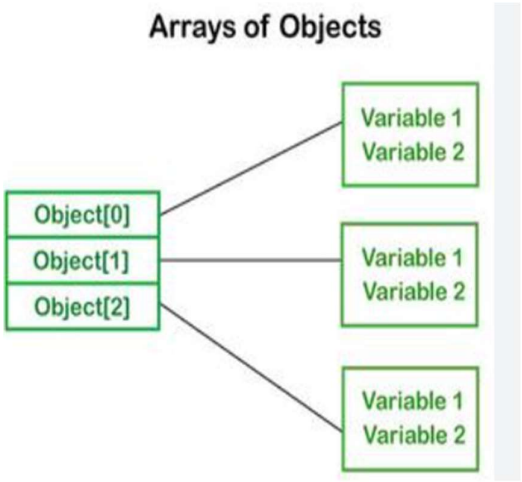
---

Books/ Journals/ Websites referred:

1. E. Balagurusamy, “Programming with Java”, McGraw-Hill.
2. E. Balagurusamy, “Object Oriented Programming with C++”, McGraw-Hill.

Pre Lab/ Prior Concepts:

Java is an object-oriented programming language. Most of the work done with the help of objects. We know that an array is a collection of the same data type that dynamically creates objects and can have elements of primitive types. Java allows us to store objects in an array. In Java, the class is also a user-defined data type. An array that conations class type elements are known as an array of objects. It stores the reference variable of the object.



Creating an Array of Objects

Before creating an array of objects, we must create an instance of the class by using the new keyword. We can use any of the following statements to create an array of objects.

Syntax:
ClassName obj[]=new ClassName[array_length]; //declare and instantiate an array of o bjects

For example:

```
class Student {
  int rno;
  String name;
  float avg;
}
Student(int r, String name, float average)
{
    rno=r;
   this.name=name;
    avg=average;
}

Student studentArray[] = new Student[n];
```

- The above statement creates the array which can hold references to n number of Student objects. It doesn't create the Student objects themselves. They have to be created separately using the constructor of the Student class. The studentArray contains n number of memory spaces in which the address of n Student objects may be stored.

```
for ( int i=0; i<studentArray.length; i++)
{
studentArray[i]=new Student(r,name,average);
}
```

- The above for loop creates n Student objects and assigns their reference to the array elements. Now, a statement like the following would be valid.

```
studentArray[i].r=1001;
.
```

Class Diagram:

Algorithm:

1. Start the program.

2. The control first goes to the main() function. It accepts the no of data to be inserted by the user and creates the no of array of Accinfo structure.

3. It asks user no of initial data to be inserted in the array of structure. If the initial size inserted is less than the size of array created than a for loop runs for initial size and user to enter the required data.

4. Now a will loop runs until user doesn't wants to exit the program and it asks user to input in from 1-4.

5. If user enters 1 than it checks whether the array is not full than it ask user to enter the required data.

6. If user enters 2 then it checks whether the array is empty or not, if not than it ask user the id to be deleted and it calls a for loop which over writes the data in that node by the preceding data.

7. If user enter 3 then it prints the total data in the array structure

8. If user enter 0 the it exits the program.

Implementation details:

```
import java.util.Scanner;


public class Customer {


   public static void main(String[] args) {


      int option;

      Scanner sc = new Scanner(System.in);


      System.out.println("Enter the number of customers:");

      int cusCount = sc.nextInt();

      acc[] arr = new acc[cusCount];


      System.out.println("Enter the initial number of accounts to be added:");

      int n = sc.nextInt();
```

```
int used_acc = n;


if (n <= cusCount) {
   for (int i = 0; i < n; i++) {
      addAccount(sc, arr, i);
   }

   do {
      System.out.println("\nChoose an option:");
      System.out.println("1. Add Account");
      System.out.println("2. Delete Account");
      System.out.println("3. Display Accounts");
      System.out.println("4. Exit");

      option = sc.nextInt();

      switch (option) {
         case 1:
            if (n >= cusCount) {
               System.out.println("No more accounts can be added.");
```

```
    } else {
        addAccount(sc, arr, n++);
    }
    break;

case 2:
    n = deleteAccount(sc, arr, n);
    break;

case 3:
    displayAccounts(arr, n);
    break;

case 4:
    System.out.println("Exiting...");
    break;

default:
    System.out.println("Invalid option. Choose between 1 and 4.");
    break;
```

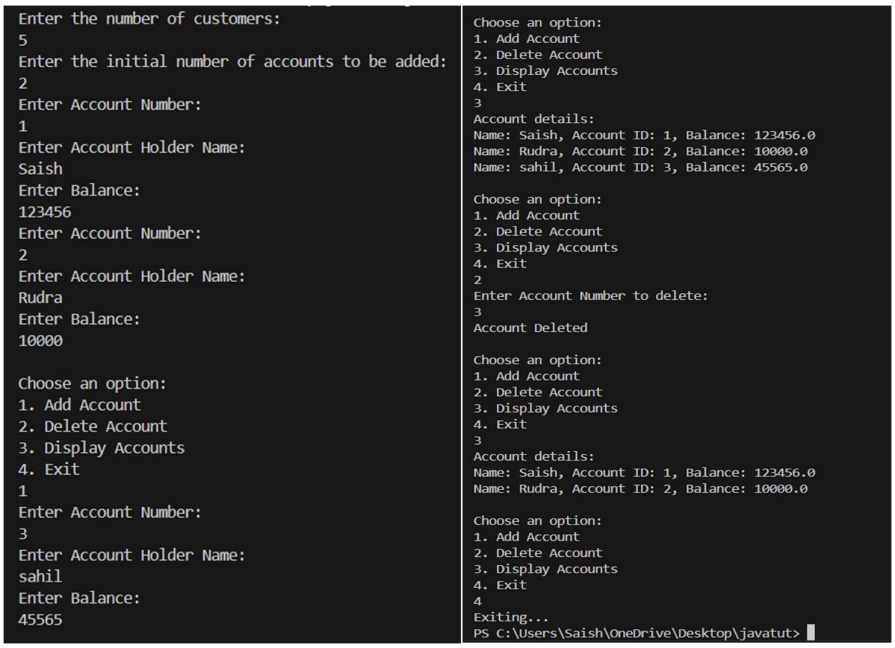
```
            }
        } while (option != 4);
    } else {
        System.out.println("Number of accounts cannot exceed number of
customers!");
    }

    sc.close();
  }

  public static void addAccount(Scanner sc, acc arr[], int index) {
    System.out.println("Enter Account Number:");
    int accNum = sc.nextInt();

    System.out.println("Enter Account Holder Name:");
    String name = sc.next();

    System.out.println("Enter Balance:");
    float bal = sc.nextFloat();
```

```
    arr[index] = new acc(accNum, name, bal);
}

public static int deleteAccount(Scanner sc, acc arr[], int n) {
    System.out.println("Enter Account Number to delete:");
    int del = sc.nextInt();
    boolean found = false;

    for (int i = 0; i < n; i++) {
        if (del == arr[i].id) {
            for (int j = i; j < n - 1; j++) {
                arr[j] = arr[j + 1];
            }
            n--;
            found = true;
            System.out.println("Account Deleted");
            break;
        }
    }
```

```
        if (!found) {
            System.out.println("Account not found.");
        }

        return n;
    }

    public static void displayAccounts(acc[] arr, int n) {
        System.out.println("Account details:");
        for (int i = 0; i < n; i++) {
            System.out.println("Name: " + arr[i].name + ", Account ID: " + arr[i].id + ",
Balance: " + arr[i].bal);
        }
    }
}

class acc {
    int id;
    String name;
    float bal;
```

```
    acc(int id, String name, float bal) {
        this.id = id;
        this.name = name;
        this.bal = bal;
    }
}
```

Output:



Conclusion:

We can use array of objects implementation and create a menu driven program

Date: ___5/9/24_____ **Signature of faculty in-charge**

Post Lab Descriptive Questions:

Q.1 If an array of objects is of size 10 and a data value have to be retrieved from 5th object then _____ syntax should be used.

a)Array_Name[4].data_variable_name;
b)Data_Type Array_Name[4].data_variable_name;
c)Array_Name[4].data_variable_name.value;
d) Array_Name[4].data_variable_name(value);
Ans:
a

Q.2 The Object array is created in _____
a)Heap memory
b) Stack memory
c) HDD
d) ROM
Ans
A

Q.3 Explain the difference between Jagged Array and Array of Object .

A jagged array consists of arrays of different sizes.
An array of objects consists of instances of a class.
Jagged arrays are used when we need to store collections where the inner collections (arrays) have variable sizes.
Arrays of objects are used when we want to store multiple objects of a class.

```
class Person {
   String name;
   int age;
   Person(String name, int age) {
      this.name = name;
      this.age = age;
   }
```

```
    void show() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}
public class Main {
    public static void main(String[] args) {
        System.out.println("Jagged Array:");
        int[][] jagged = new int[3][];
        jagged[0] = new int[] {1, 2, 3};
        jagged[1] = new int[] {4, 5};
        jagged[2] = new int[] {6, 7, 8, 9};
        for (int i = 0; i < jagged.length; i++) {
            for (int j = 0; j < jagged[i].length; j++) {
                System.out.print(jagged[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println("\nArray of Objects:");
        Person[] people = new Person[3];
        people[0] = new Person("Saish", 19);
        people[1] = new Person("Rudra", 19);
        people[2] = new Person("Sahil", 19);
        for (Person p : people) {
            p.show();
        }
    }
}
```

Output:

```
Jagged Array:
1 2 3
4 5
6 7 8 9

Array of Objects:
Name: Saish, Age: 19
Name: Rudra, Age: 19
Name: Sahil, Age: 19
```