Batch: B2	**Roll No.: 16010124107**
Experiment / assignment / tutorial No. 5	
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 the 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

---

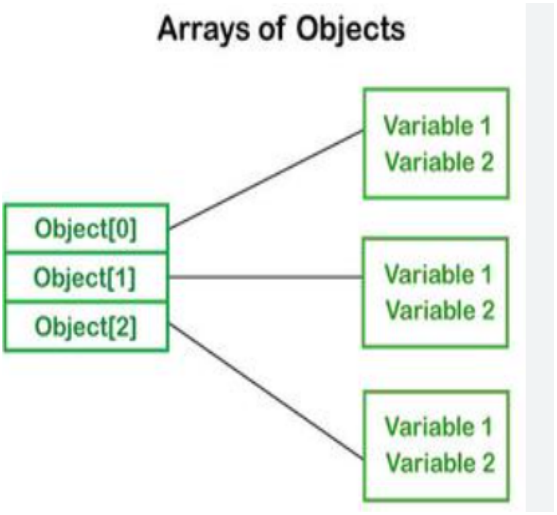
Books/ Journals/ Websites referred:

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 is 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

contains 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.

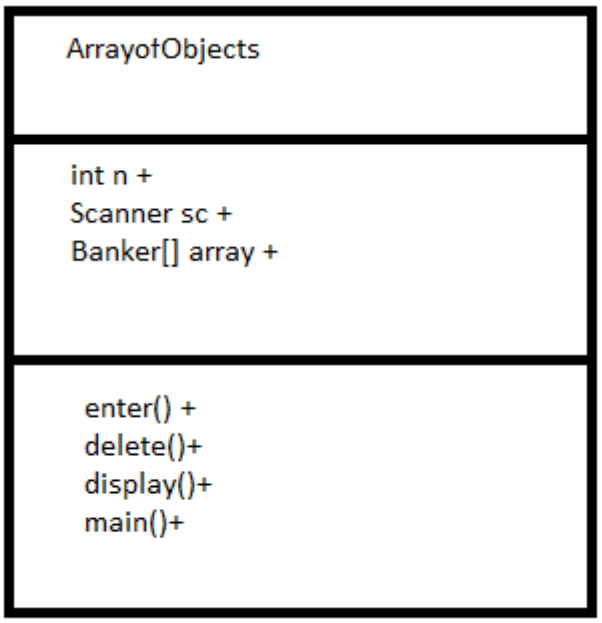
```
for ( int i=0; i<studentArray.length; i++)
{
studentArray[i]=new Student(r,name,average);
}
```

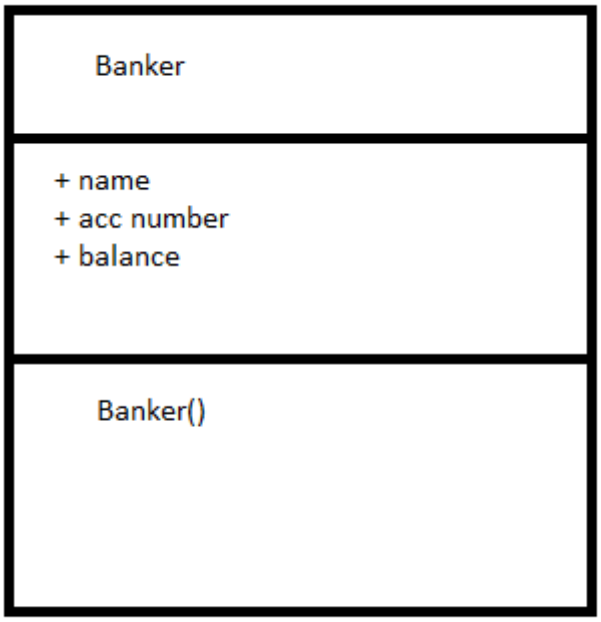
- The above for loop creates n Student objects and assigns their reference to the array elements. Now, a statement like the following would be valid.

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

.

Class Diagram:





Algorithm:

1. Start
2. Initialize an empty list of accounts
3. Repeat until user exits:
4. Show menu and get user choice
5. If choice is Add account, get account details and add to list
6. If choice is Delete account, get account number and remove it from list if found
7. If choice is Display accounts, show all account details
8. If choice is anything else, exit
9. End

Implementation details:

```
import java.util.*;


class Banker{
```

```
    String name;
    long acc;
    double balance;

    Banker(String n, long a, double b)
    {
        name = n;
        acc = a;
        balance = b;
    }
}

class arrayofobjects{

    static Scanner sc = new Scanner(System.in);
    int n = 0;
    Banker[] array = new Banker[n];

    arrayofobjects() {
        array = new Banker[0];
    }

    public void enter()
```

```
{
    System.out.println("Enter number of users");
    n = sc.nextInt();
    array = new Banker[n];
    for(int i=0;i<n;i++)
    {
        System.out.println("Enter name");
        String s = sc.next();
        System.out.println("Enter account number");
        long ac = sc.nextLong();
        System.out.println("Enter balance");
        double d = sc.nextDouble();
        array[i] = new Banker(s,ac,d);
    }
}

public void Delete(long acc)
{
    for(int i=0;i<n;i++)
    {
        if(array[i].acc == acc)
        {
            for(int j=i+1;j<n;j++)
```

```
                {
                    array[j-1] = array[j];
                }
                n--;
                return;
            }
        }
        System.out.println("Not found");
    }

    public void Display()
    {
        for(int i = 0; i < n; i++) {
            Banker x = array[i];
            System.out.println("Name: "+x.name);
            System.out.println("Account Number: "+x.acc);
            System.out.println("Balance: "+x.balance);
        }
    }

    public static void main(String args[])
    {
        arrayofobjects obj = new arrayofobjects();
```

```
        while(true){

            System.out.println("Choice:\n1.Add  account\n2.Delete  account
details\n3.Display account details\nPress Any Number To Exit");

       int choice = sc.nextInt();

       switch(choice)

       {

        case 1:

        obj.enter();

        break;

        case 2:

        System.out.println("Enter acc number to delete\n");

        long an = sc.nextLong();

        obj.Delete(an);

        break;

        case 3:

        obj.Display();

        break;

        default:

        return;

       }

    }

    }

}
```

Output:

```
Choice:
1.Add account
2.Delete account details
3.Display account details
Press Any Number To Exit
1
Enter number of users
3
Enter name
riya 12
Enter account number
Enter balance
1232
Enter name
ashwera
Enter account number
789
Enter balance
0923
Enter name
kiara
Enter account number
4567
Enter balance
231
Choice:
1.Add account
2.Delete account details
3.Display account details
Press Any Number To Exit
```

PROBLEMS OUTPUT DEBUG CONSOLE TERM

```
Account Number: 789
Balance: 923.0
Name: kiara
Account Number: 4567
Balance: 231.0
Choice:
1.Add account
2.Delete account details
3.Display account details
Press Any Number To Exit
2
Enter acc number to delete

12
Choice:
1.Add account
2.Delete account details
3.Display account details
Press Any Number To Exit
3
Name: ashwera
Account Number: 789
Balance: 923.0
Name: ashwera
Account Number: 789
Balance: 923.0
Name: kiara
Account Number: 4567
Name: ashwera
Account Number: 789
Balance: 923.0
Name: kiara
Name: ashwera
Account Number: 789
Balance: 923.0
Account Number: 789
Balance: 923.0
Balance: 923.0
Name: kiara
Account Number: 4567
Balance: 231.0
Choice:
1.Add account
2.Delete account details
3.Display account details
```

Conclusion:

An array of objects is a collection of objects with various attributes that helps us create instances of objects in a single class. For example, in a bank, there will be multiple users and each user can be a member of the array. Now each member of the bank will have some attributes which can also be stored in the database using the array of objects system. this also helps encapsulate the properties of one object together.

Date:01/09/2025 **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)Array_Name[4].data_variable_name;

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

Q.3 Explain the difference between Jagged Array and Array of Objects .
Ans:
A jagged array is a multi dimensional array that can hold a variable number of rows and columns for each dimension. For example, a double dimensional jagged array can have a fixed number of rows and varying number of columns for each row.

On the other hand, an array of objects can be single or multi dimensional but the data type of this array is an Object. Whatever object is stored in this array of objects, its all attributes are associated with the array and so on.