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

Batch: E2	**Roll No.: 16010123325**
Experiment / assignment / tutorial No.	
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.

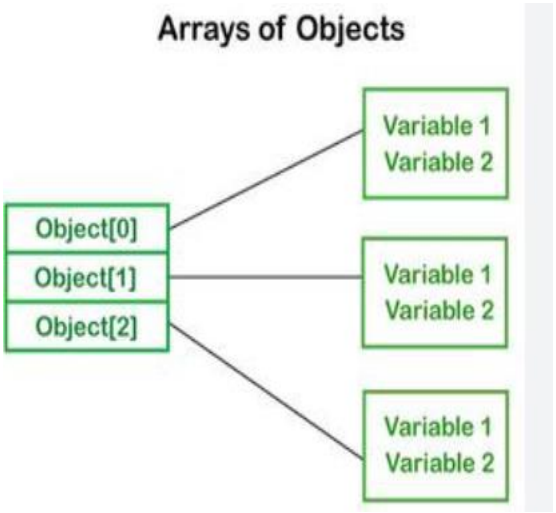
Expected OUTCOME of Experiment:
CO1: Apply the features of object oriented programming languages. (C++ and Java)
CO2:Explore arrays, vectors, classes and objects in C++ and Java

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.

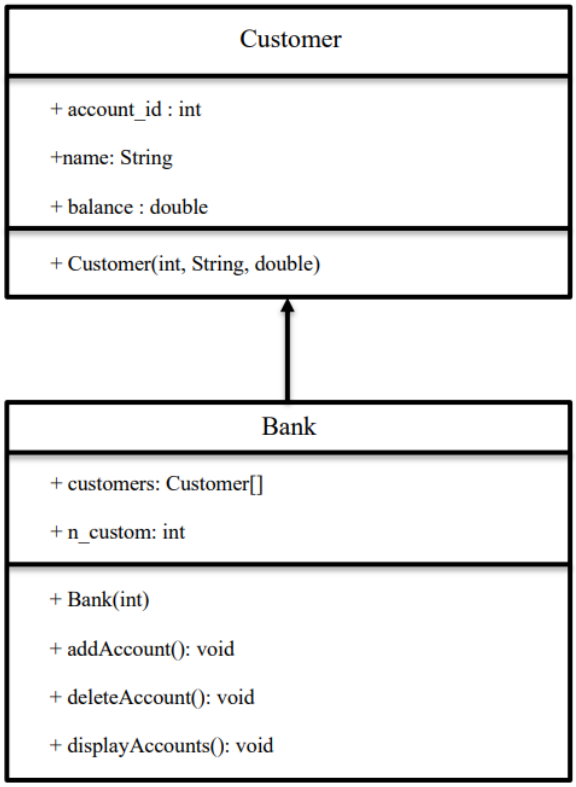
```
for ( int i=0; i<studentArray.length; i++)
{
studentArray[i]=new Student(r,name,average);
}
```

- 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. Initialize the Program

- Prompt the user to enter the maximum number of accounts
- Create a `Bank` object with an array of `Customer` objects and set the initial account count (`n_custom`) to 0

2. Display Menu and Process Choices

- Show menu options: Add account, Delete account, Display accounts, Exit.
- Read user input to determine action.

3. Menu Actions

- Add Account:
  - Prompt for `account_id`, `name`, and `balance`.
  - Create a new `Customer` object and add it to the `customers` array if there is space.
  - Increment `n_custom`.

- Delete Account:
  - Prompt for the `account_id` to delete.
  - Search for the `account_id` in the `customers` array.
  - If found, shift subsequent elements left to overwrite the deleted account and
  - decrement `n_custom`.
  - If not found, display an error message

- Display Accounts:
  - Loop through the `customers` array up to `n_custom` and print each customer's details (`account_id`, `name`, `balance`)
  - Exit

4. Repeat

- Return to the menu until the user selects "Exit".

Implementation details:

```
import java.util.Scanner;

class Customer {
    int accountId;
    String name;
    double balance;

    public Customer(int accountId, String name, double balance) {
        this.accountId = accountId;
        this.name = name;
        this.balance = balance;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of customers:");
        int n = scanner.nextInt();
        Customer[] customers = new Customer[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter account id, name and balance for
customer " + (i + 1));
            int accountId = scanner.nextInt();
            String name = scanner.next();
            double balance = scanner.nextDouble();
            customers[i] = new Customer(accountId, name, balance);
        }

        while (true) {
            System.out.println("1. Add account\n2. Delete account\n3.
Display account details\n4. Exit");
            System.out.println("Enter your choice:");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    customers = addAcc(customers, scanner);
                    break;
```

```
                case 2:
                    customers = deleteAcc(customers, scanner);
                    break;
                case 3:
                    display(customers);
                    break;
                case 4:
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice. Please try
again.");
            }
        }
    }

    private static Customer[] addAcc(Customer[] customers, Scanner
scanner) {
        System.out.println("Enter account id, name and balance for new
customer:");
        int accountId = scanner.nextInt();
        String name = scanner.next();
        double balance = scanner.nextDouble();
        Customer newCustomer = new Customer(accountId, name, balance);
        Customer[] newCustomers = new Customer[customers.length + 1];
        System.arraycopy(customers, 0, newCustomers, 0,
customers.length);
        newCustomers[customers.length] = newCustomer;
        return newCustomers;
    }

    private static Customer[] deleteAcc(Customer[] customers, Scanner
scanner) {
        System.out.println("Enter account id to delete:");
        int accountId = scanner.nextInt();
        boolean found = false;
        int index = -1;
        for (int i = 0; i < customers.length; i++) {
            if (customers[i].accountId == accountId) {
                found = true;
                index = i;
                break;
```

```
            }
        }
        if (found) {
            Customer[] newCustomers = new Customer[customers.length - 1];
            System.arraycopy(customers, 0, newCustomers, 0, index);
            System.arraycopy(customers, index + 1, newCustomers, index,
customers.length - index - 1);
            return newCustomers;
        } else {
            System.out.println("Account not found.");
            return customers;
        }
    }

    private static void display(Customer[] customers) {
        for (int i = 0; i < customers.length; i++) {
            System.out.println("Account Id: " + customers[i].accountId +
", Name: " + customers[i].name + ", Balance: " + customers[i].balance);
        }
    }
}
```

Output:

```
java -cp /tmp/DXG1yPvAh8/Main
Enter the number of customers:
2
Enter account id, name and balance for customer 1
1
Shrey
999
Enter account id, name and balance for customer 2
2
Damn
777
1. Add account
2. Delete account
3. Display account details
4. Exit
Enter your choice:
3
Account Id: 1, Name: Shrey, Balance: 999.0
Account Id: 2, Name: Damn, Balance: 777.0
```

```
1. Add account
2. Delete account
3. Display account details
4. Exit
Enter your choice:
1
Enter account id, name and balance for new customer:
3
Jeez
1000
1. Add account
2. Delete account
3. Display account details
4. Exit
Enter your choice:
3
Account Id: 1, Name: Shrey, Balance: 999.0
Account Id: 2, Name: Damn, Balance: 777.0
Account Id: 3, Name: Jeez, Balance: 1000.0
```

```
1. Add account
2. Delete account
3. Display account details
4. Exit
Enter your choice:2

2
Enter account id to delete:
2
1. Add account
2. Delete account
3. Display account details
4. Exit
Enter your choice:
3
Account Id: 1, Name: Shrey, Balance: 999.0
Account Id: 3, Name: Jeez, Balance: 1000.0
```

Conclusion:

The above program highlights use of array objects in a bank management system, with options to add, delete, display accounts.

Date: _____ **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 .

Jagged Array: A jagged array is an array whose elements are also arrays, and these inner arrays can have different lengths. This allows for a flexible structure where each row or element can have a varying number of columns or values. It's commonly used when dealing with data that naturally has an irregular shape, like a list of students where each student has a different number of grades.

Array of Objects: An array of objects is an array where each element is an object reference of the same class type. This array allows storing multiple instances of a class, enabling easy access to each object's properties and methods. It's useful when you need to manage a collection of objects, like a list of employees where each object represents an individual employee with various attributes such as name, age, and salary.