Batch: A4 Roll No.: 16010122083

Experiment / assignment / tutorial No. 04

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,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: Understand the features of object oriented programming compared with procedural approach with C++ and Java

CO2: Explore arrays, vectors, classes and objects in C++ and Java.

Books/ Journals/ Websites referred:

1. Ralph Bravaco , Shai Simoson , “Java Programing From the Group Up” Tata McGraw-Hill.
2. Grady Booch, Object Oriented Analysis and Design .

Pre Lab/ Prior Concepts:

Arrays of Objects:

Unlike traditional array which store values like string, integer, boolean, etc. array of objects stores objects. The array elements store the location of reference variables of the object.

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;
```

.

Algorithm:

1. Scanner class is imported from the util package.
2. A Parent Class is created by name Account which displays the details of the person
3. Created two Subclass name Current_Account and Saving Account which stores the value of amount and calculate interest differently with respective rate of interest given by the user.
4. These classes above are extended by the Parent Class Account
5. A main function is created which creates a Objects of the subclass where the value is accepted and then displayed by call the Sub Classes function and displaying it by the Parent Class , Account.

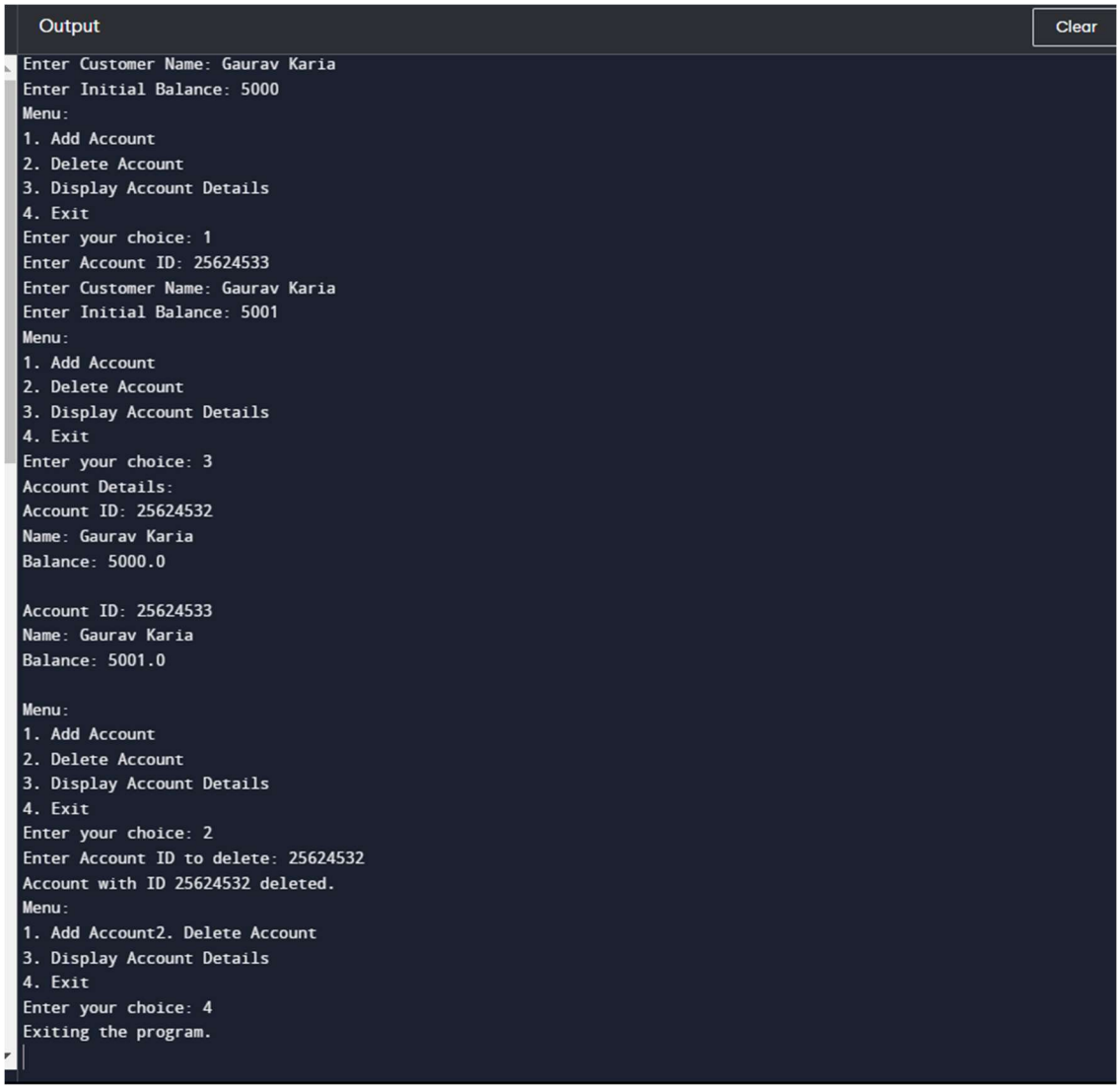
Implementation details:

```
import java.util.Scanner;

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

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

public class CustomerAccountManagement {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = 10; // Maximum number of customers
        Customer[] customers = new Customer[n];
        int count = 0; // Current number of customers in the array

        while (true) {
            System.out.println("Menu:");
            System.out.println("1. Add Account");
            System.out.println("2. Delete Account");
            System.out.println("3. Display Account Details");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();

            switch (choice)
            {
                case 1:
                    if (count < n)
                    {
                        System.out.print("Enter Account ID: ");
                        int account_id = scanner.nextInt();
                        System.out.print("Enter Customer Name: ");
                        scanner.nextLine(); // Consume the newline
character
                        String name = scanner.nextLine();
```

```
                        System.out.print("Enter Initial Balance: ");
                        double balance = scanner.nextDouble();

                        customers[count] = new Customer(account_id, name,
balance);
                        count++;
                    }
                    else
                    {
                        System.out.println("Cannot add more accounts.
Array is full.");
                    }
                    break;

                case 2:
                    System.out.print("Enter Account ID to delete: ");
                    int idToDelete = scanner.nextInt();
                    boolean found = false;

                    for (int i = 0; i < count; i++)
                    {
                        if (customers[i].account_id == idToDelete)
                        {
                            // Shift elements to fill the gap
                            for (int j = i; j < count - 1; j++)
                            {
                                customers[j] = customers[j + 1];
                            }
                            count--;
                            found = true;
                            System.out.println("Account with ID " +
idToDelete + " deleted.");
                            break;
                        }
                    }

                    if (!found) {
                        System.out.println("Account with ID " +
idToDelete + " not found.");
                    }
                    break;

                case 3:
                    System.out.println("Account Details:");
```

```
                    for (int i = 0; i < count; i++)
                    {
                        System.out.println("Account ID: " +
customers[i].account_id);
                        System.out.println("Name: " + customers[i].name);
                        System.out.println("Balance: " +
customers[i].balance);
                        System.out.println();
                    }
                    break;

                case 4:
                    System.out.println("Exiting the program.");
                    scanner.close();
                    System.exit(0);
                    break;

                default:
                    System.out.println("Invalid choice. Please try
again.");
            }
        }
    }
}
```

Output:



Conclusion: We learnt the concept of inheritance and the implementation of the it with a example

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:
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: Heap Memory