Batch: A3	**Roll No.: 16010121045**
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;

.

Class Diagram:

Class name	Exp4
Variables	-
Functions	main()

Class name	acccount
Variables	Int id, String name, double balance
Functions	-

Algorithm:

1. Create a class Account with attributes int is, String name and float Balance.
2. Create a constructor for this class.
3. Create a public class Expt_4.
4. Define the main method in this class.
5. Create an object of Scanner class.
6. Get the total number of customers from the user and declare the array of same length.
7. Define a while loop.
8. In this while loop get choice from the user to Add, Delete, Display the account or to Exit.
9. If user selects option 1.
10. Get the number of accounts to be added initially.
11. Get account number, account holder name and balance.
12. Store this in the array of objects defines earlier.
13. If user selects option 2.
14. Get the account number from the user.
15. Find that account number in the array and shift the next element of the array at that position.
16. If user selects option 3.
17. Print the contents of array using for loop.
18. If user selects option 4.
19. Exit

Implementation details:

```
import java.util.Scanner;

class exp4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of customers: ");
        int n = sc.nextInt();
        account arr[] = new account[n];
        int t = 0;
        while (t != 4) {
            System.out.println("(1) Add Account");
            System.out.println("(2) Delete Account");
            System.out.println("(3) Display Accounts");
            System.out.println("(4) Exit");
            t = sc.nextInt();
            if (t == 1)
                addAccount(arr, n);
            else if (t == 2) {
                System.out.print("Enter Account ID. to
Delete: ");
                int id = sc.nextInt();
                deleteAccount(arr, id, n);
            } else if (t == 3){
                System.out.println("Enter the account id to
search: ");
                int id=sc.nextInt();
                displayAccount(arr, id,n);
            }
            else if (t == 4)
                break;
            else
                System.out.println("Please Enter Correct
Option!");

        }
    }
```

```
    static void addAccount(account arr[], int n) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Account ID: ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Account Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Account Balance: ");
        double balance = sc.nextDouble();
        boolean check = false;
        for (int i = 0; i < n; i++) {
            if(arr[i]!=null){
                if(arr[i].id==id){
                check=true;
                System.out.println("Cannot add the above
account id");
                break;
                }
            }
            else {
                arr[i] = new account(id, name, balance);
                check = true;
                break;
            }
        }
        if (check == false)
            System.out.println("There is not enough space in
Array!");
    }

    static void deleteAccount(account arr[], int id, int n) {
        boolean check = false;
        for (int i = 0; i < n && arr[i]!=null; i++) {
            if (arr[i].id == id) {
                check = true;
                for (int j = i; j < n - 1; j++)
                    arr[j] = arr[j + 1];
                arr[n - 1] = null;
            }
        }
```

```
        if (check == false)
            System.out.println("Account ID not found!");
    }

    static void displayAccount(account arr[], int id,int n) {
        boolean check=false;
        for(int i=0;i<n && arr[i]!=null;i++){
            if(arr[i].id==id){
            System.out.println("\nAccount id: " + arr[i].id);
            System.out.println("Account name: " +
arr[i].name);
            System.out.println("Account balance: " +
arr[i].balance + "\n");
            check=true;
            break;
            }
        }

        if(check==true)
            System.out.println("Account Not Found!");
    }
}

class account {
    int id;
    String name;
    double balance;

    account() {
        id = 0;
        name = null;
        balance = 0;
    }

    account(int pid, String pname, double pbalance) {
        id = pid;
        name = pname;
        balance = pbalance;
    }
}
```

Output:

```
Enter the number of customers: 3
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
1
Enter Account ID: 1
Enter Account Name: Pargat
Enter Account Balance: 100000
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
1
Enter Account ID: 2
Enter Account Name: Vishrut
Enter Account Balance: 2000
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
1
Enter Account ID: 3
Enter Account Name: Meet
Enter Account Balance: 3000
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
3
Enter the account id to search:
1

Account id: 1
Account name: Pargat
Account balance: 100000.0
```

```
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
1
Enter Account ID: 3
Enter Account Name: Meet
Enter Account Balance: 3000
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
3
Enter the account id to search:
1

Account id: 1
Account name: Pargat
Account balance: 100000.0

Account Not Found!
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
2
Enter Account ID. to Delete: 2
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
3
Enter the account id to search:
2
(1) Add Account
(2) Delete Account
(3) Display Accounts
(4) Exit
4
pargat@Router Exp4 % `
```

Conclusion: The experiment was executed successfully.

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