**EESHANYA JOSHI**

**Batch: A3 Roll No.: 16010122074**

**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 McGrawHill.
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:**

|  |
| --- |
| **Account** |
| accountID : Int  name: String  balance: Double  Scanner sc |
| +printAccountData() |

|  |
| --- |
| **Question 4** |
| n : Int  choice: Int  accountID: Int  Customername: String  balance: Double  deleteID :Int  Scanner sc |
| +main() |

**Algorithm:**

1. Import necessary libraries (java.util.ArrayList and java.util.Scanner).

2. Define a Customer class with private fields id and name, along with a constructor to initialize them.

3. Define an Account class with private fields accountNumber, balance, and an ArrayList of Customer objects to represent account holders. Create a constructor to initialize these fields.

4. Implement methods in the Account class:

* getAccountNumber(): Returns the account number.
* addAccountHolder(Customer customer): Adds a customer to the account's list of account holders.
* removeAccountHolder(Customer customer): Removes a customer from the account's list of account holders.
* getBalance(): Returns the current balance.
* deposit(double amount): Deposits a specified amount into the account if it's a positive value.
* withdraw(double amount): Withdraws a specified amount from the account if the balance is sufficient and the amount is positive.
* displayAccountDetails(): Displays account information, including the account number, balance, and account holders.

5. Define the Main class with the main method as the program's entry point.

6. Inside the main method:

* Create a Scanner object to read user input.
* Create an ArrayList called accounts to store instances of the Account class.
* Start an infinite loop to display a menu for various banking operations.

Within the loop, use a switch statement to handle user choices:

**Option 1: Open Account**

* Prompt the user for account number and initial balance.
* Prompt for the number of account holders and their details (ID and name).
* Create an Account object and add it to the accounts list.

**Option 2: Close Account**

* Prompt the user for the account number to close.
* Search for the account in the accounts list and remove it if found.

**Option 3: Deposit**

* Prompt the user for the account number and deposit amount.
* Find the account and call the deposit method on it.

**Option 4: Withdraw**

* Prompt the user for the account number and withdrawal amount.
* Find the account and call the withdraw method on it.

**Option 5: Display Account Details**

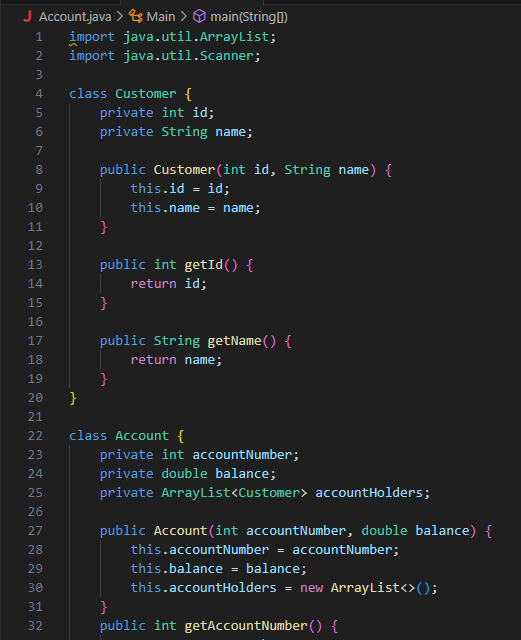
* Prompt the user for the account number.
* Find the account and call the displayAccountDetails method on it.

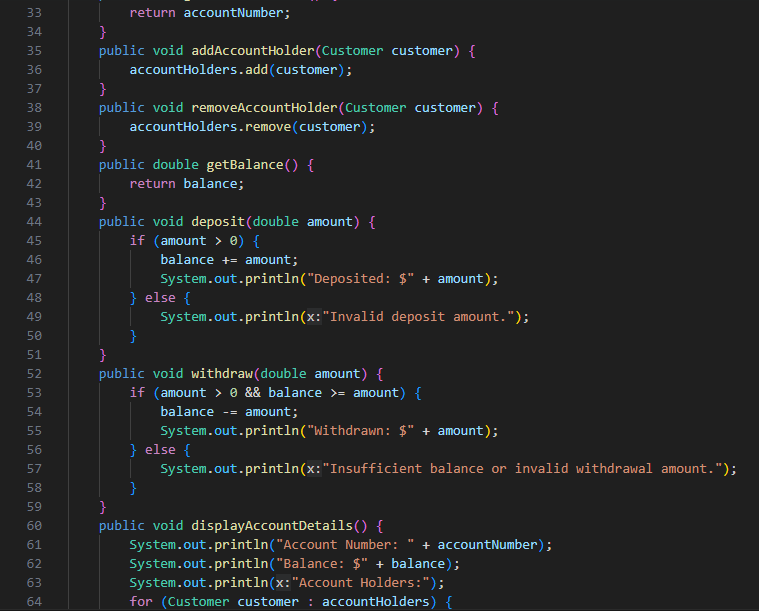
**Option 6: Exit**

Close the Scanner and exit the program

Default: Display an error message for invalid choices.

**Implementation details:**

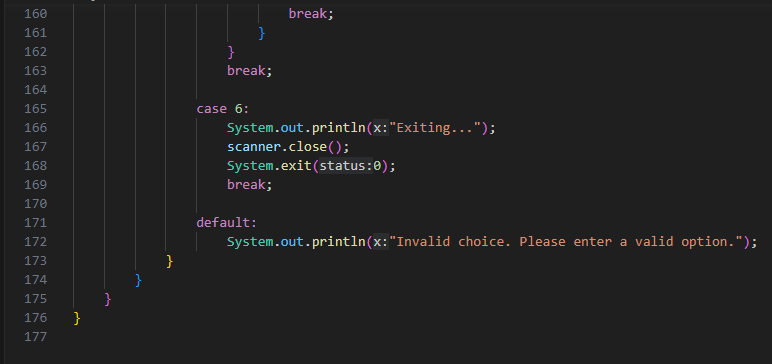
****

****

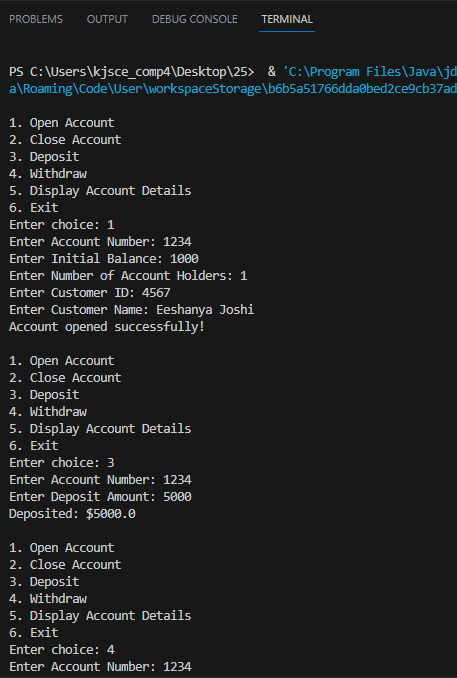
****

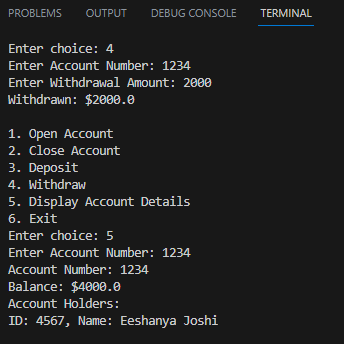
****

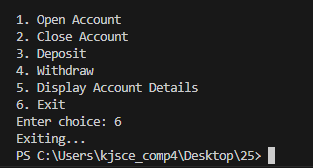
****

****

**OUTPUT**







**Conclusion:** With the help of array of projects bank account is created and using switch case add, delete and display of account is used using array.

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty incharge**

**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)**