

**K.R. Mangalam University School of Engineering &  
Technology**

## **Fundamentals Of Java Programming**

### **Assignment 1**

#### **Banking Application for Accounts Management**

Submitted by: SOPHIA SONI

Roll No: 2401201042

Course: BCA (AI & DS)

CODE-

```
1 import java.util.Scanner;
2
3 // Account Class Definition
4 class Account {
5     // fields
6     private final int accountNumber;
7     private String name, email, phone;
8     private double balance;
9
10    // constructor
11    Account(int accNo, String name, double balance, String email, String phone) {
12        this.accountNumber = accNo;
13        this.name = name;
14        this.balance = balance;
15        this.email = email;
16        this.phone = phone;
17    }
18
19    // method to get account number
20    public int getAccountNumber() {
21        return accountNumber;
22    }
23
24    // method to deposit money
25    void deposit(double amt) {
26        if (amt > 0) {
27            balance += amt;
28            System.out.println("Deposit Successful. Balance: " + balance);
29        } else {
30            System.out.println("Invalid deposit amount.");
31        }
32    }

```

```
33
34     // method to withdraw money
35     void withdraw(double amt) {
36         if (amt > 0 && amt <= balance) {
37             balance -= amt;
38             System.out.println("Withdrawal Successful. Balance: " + balance);
39         } else {
40             System.out.println("Invalid or insufficient balance.");
41         }
42     }
43
44     // method to display account details
45     void show() {
46         System.out.println("Account Number: " + accountNumber + " | Name: " + name +
47                             " | Balance: " + balance + " | Email: " + email +
48                             " | Phone: " + phone);
49     }
50
51     // method to update contact details
52     void update(String email, String phone) {
53         this.email = email;
54         this.phone = phone;
55         System.out.println("Contact updated!");
56     }
57 }
58
59 // Main Banking Application Class
60 public class BankingApp {
61 }
```

```

62     // Static members for the application state
63     static Scanner sc = new Scanner(System.in);
64     static Account[] accounts = new Account[100]; // Array to store up to 100 accounts
65     static int count = 0; // Counter for the number of accounts created
66
67     // Method to find an account by account number
68     static Account find(int accNo) {
69         for (int i = 0; i < count; i++) {
70             if (accounts[i].getAccountNumber() == accNo) {
71                 return accounts[i];
72             }
73         }
74         return null; // Return null if not found
75     }
76
77     Run | Debug
78     public static void main(String[] args) {
79
80         while (true) {
81             // Display Menu (Customized for Sophia)
82             System.out.println(x: "\n*** WELCOME TO SOPHIA'S BANKING APP ***");
83             System.out.println(x: "1. Create Account");
84             System.out.println(x: "2. Deposit");
85             System.out.println(x: "3. Withdraw");
86             System.out.println(x: "4. Show Account Details");
87             System.out.println(x: "5. Update Contact Info");
88             System.out.println(x: "6. Exit");
89             System.out.print(s: "Enter your choice: ");
90
91             int choice = sc.nextInt();
92             sc.nextLine(); // consume newline
93
94             switch (choice) {
95                 case 1 -> {
96                     System.out.print(s: "Enter Name: ");
97                     String name = sc.nextLine();
98                     System.out.print(s: "Enter Initial Deposit: ");
99                     double bal = sc.nextDouble();
100                    sc.nextLine();
101                    System.out.print(s: "Enter Email: ");
102                    String email = sc.nextLine();
103                    System.out.print(s: "Enter Phone: ");
104                    String phone = sc.nextLine();
105
106                    accounts[count] = new Account(1000 + count + 1, name, bal, email, phone);
107                    System.out.println("Account Created: " + accounts[count].getAccountNumber());
108                    count++;
109                }
110
111                case 2 -> {
112                    System.out.print(s: "Acc No: ");
113                    int accNo = sc.nextInt();
114                    System.out.print(s: "Amount to deposit: ");
115                    double amt = sc.nextDouble();
116
117                    Account a = find(accNo);
118                    if (a != null) {
119                        a.deposit(amt);
120                    } else {
121                        System.out.println(x: "Not found.");
122                    }
123                }
124            }
125        }
126    }

```

Activate Window  
Go to Settings to

```
121 }  
122 }  
123  
124 case 3 -> {  
125     System.out.print(s: "Acc No: ");  
126     int accNo = sc.nextInt();  
127     System.out.print(s: "Amount to withdraw: ");  
128     double amt = sc.nextDouble();  
129  
130     Account a = find(accNo);  
131     if (a != null) {  
132         a.withdraw(amt);  
133     } else {  
134         System.out.println(x: "Not found.");  
135     }  
136 }  
137  
138 case 4 -> {  
139     System.out.print(s: "Acc No: ");  
140     int accNo = sc.nextInt();  
141     Account a = find(accNo);  
142     if (a != null) {  
143         a.show();  
144     } else {  
145         System.out.println(x: "Not found.");  
146     }  
147 }  
148  
149 case 5 -> {  
150     Systerm.out.print(s: "Acc No: ");  
151  
152     int accNo = sc.nextInt();  
153     sc.nextLine();  
154     System.out.print(s: "New Email: ");  
155     String email = sc.nextLine();  
156     System.out.print(s: "New Phone: ");  
157     String phone = sc.nextLine();  
158  
159     Account a = find(accNo);  
160     if (a != null) {  
161         a.update(email, phone);  
162     } else {  
163         System.out.println(x: "Not found.");  
164     }  
165  
166 case 6 -> {  
167     System.out.println(x: "Exiting. Thank you!");  
168     return;  
169 }  
170  
171 default -> System.out.println(x: "Invalid choice!");  
172 }
```

## **OUTPUT:**

```
*** WELCOME TO SOPHIA'S BANKING APP ***
1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit
Enter your choice: 1
Enter Name: SOPHIA
Enter Initial Deposit: 5000
Enter Email: Sophia@gmail.com
Enter Phone: 8630897370
Account Created: 1001

*** WELCOME TO Sophia's BANKING APP ***
1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit
Enter your choice: 2
Acc No: 1001
Amount to deposit: 2000
Deposit Successful. Balance: 7000.0
```