

# ASSIGNMENT-1

**Name-Nandini Kumari**

**Roll no-2401201085**

**Course-BCA(AI&DS)**

**Input-**

```
3 import java.util.Scanner;
4
5 class Account {
6     private int accountNumber;
7     private String accountHolderName;
8     private double balance;
9     private String email;
10    private String phoneNumber;
11
12    public Account(int accountNumber, String accountHolderName, double balance, String email, String phoneNumber) {
13        this.accountNumber = accountNumber;
14        this.accountHolderName = accountHolderName;
15        this.balance = balance;
16        this.email = email;
17        this.phoneNumber = phoneNumber;
18    }
19
20    public int getAccountNumber() {
21        return accountNumber;
22    }
23    public void deposit(double amount) {
24        if (amount > 0) {
25            balance += amount;
26            System.out.println("Amount deposited successfully!");
27        } else {
28            System.out.println("Invalid amount. Deposit must be positive.");
29        }
30    }
31}
```

```
29     }
30
31     public void withdraw(double amount) {
32         if (amount > 0) {
33             if (amount <= balance) {
34                 balance -= amount;
35                 System.out.println("Withdrawal successful!");
36             } else {
37                 System.out.println("Insufficient balance!");
38             }
39         } else {
40             System.out.println("Invalid amount. Withdrawal must be positive.");
41         }
42     }
43
44     public void displayAccountDetails() {
45         System.out.println("\n--- Account Details ---");
46         System.out.println("Account Number : " + accountNumber);
47         System.out.println("Account Holder : " + accountHolderName);
48         System.out.println("Balance      : " + balance);
49         System.out.println("Email        : " + email);
50         System.out.println("Phone Number : " + phoneNumber);
51         System.out.println("-----\n");
52     }
53
54     public void updateContactDetails(String email, String phoneNumber) {
55         this.email = email;
56         this.phoneNumber = phoneNumber;
57         System.out.println("Contact details updated successfully!");
58     }
59 }
60
61 class UserInterface {
62     private Account[] accounts = new Account[100];
63     private int count = 0;
64     private Scanner sc = new Scanner(System.in);
65
66     private Account findAccount(int accNumber) {
67         for (int i = 0; i < count; i++) {
68             if (accounts[i].getAccountNumber() == accNumber) {
69                 return accounts[i];
70             }
71         }
72         return null;
73     }
74
75     public void createAccount() {
76         System.out.print("Enter account holder name: ");
77         String name = sc.nextLine();
78
79         System.out.print("Enter initial deposit amount: ");
80         double amount = sc.nextDouble();
```

```
81.     sc.nextLine();
82.
83.     System.out.print(s: "Enter email address: ");
84.     String email = sc.nextLine();
85.
86.     System.out.print(s: "Enter phone number: ");
87.     String phone = sc.nextLine();
88.
89.     int accNo = 1000 + count + 1;
90.
91.     accounts[count] = new Account(accNo, name, amount, email, phone);
92.     count++;
93.
94.     System.out.println("Account created successfully with Account Number: " + accNo);
95. }
96.
97. public void performDeposit() {
98.     System.out.print(s: "Enter account number: ");
99.     int acc = sc.nextInt();
100.    System.out.print(s: "Enter amount to deposit: ");
101.    double amt = sc.nextDouble();
102.
103.    Account a = findAccount(acc);
104.    if (a != null) a.deposit(amt);
105.    else System.out.println(x: "Account not found!");
106. }
```

```
108. public void performWithdrawal() {
109.     System.out.print(s: "Enter account number: ");
110.     int acc = sc.nextInt();
111.     System.out.print(s: "Enter amount to withdraw: ");
112.     double amt = sc.nextDouble();
113.
114.     Account a = findAccount(acc);
115.     if (a != null) a.withdraw(amt);
116.     else System.out.println(x: "Account not found!");
117. }
118.
119. public void showAccountDetails() {
120.     System.out.print(s: "Enter account number: ");
121.     int acc = sc.nextInt();
122.
123.     Account a = findAccount(acc);
124.     if (a != null) a.displayAccountDetails();
125.     else System.out.println(x: "Account not found!");
126. }
127.
128. public void updateContact() {
129.     System.out.print(s: "Enter account number: ");
130.     int acc = sc.nextInt();
131.     sc.nextLine();
132.
133.     Account a = findAccount(acc);
134. }
```

```
135     if (a != null) {
136         System.out.print(s: "Enter new email address: ");
137         String email = sc.nextLine();
138
139         System.out.print(s: "Enter new phone number: ");
140         String phone = sc.nextLine();
141
142         a.updateContactDetails(email, phone);
143     } else {
144         System.out.println(x: "Account not found!");
145     }
146 }
147
148 public void mainMenu() {
149     while (true) {
150         System.out.println(x: "\n--- Welcome to the Banking Application ---");
151         System.out.println(x: "1. Create a new account");
152         System.out.println(x: "2. Deposit money");
153         System.out.println(x: "3. Withdraw money");
154         System.out.println(x: "4. View account details");
155         System.out.println(x: "5. Update contact details");
156         System.out.println(x: "6. Exit");
157         System.out.print(s: "Enter your choice: ");
158
159         int choice = sc.nextInt();
160         sc.nextLine();
161
162         switch (choice) {
163             case 1: createAccount(); break;
164             case 2: performDeposit(); break;
165             case 3: performWithdrawal(); break;
166             case 4: showAccountDetails(); break;
167             case 5: updateContact(); break;
168             case 6:
169                 System.out.println(x: "Thank you for using the Banking Application!");
170                 return;
171             default:
172                 System.out.println(x: "Invalid choice! Please try again.");
173         }
174     }
175 }
176
177 Run|Debug
178 public static void main(String[] args) {
179     UserInterface ui = new UserInterface();
180     ui.mainMenu();
181 }
182 }
```

## Output-

```
--- Welcome to the Banking Application ---
1. Create a new account
2. Deposit money
3. Withdraw money
4. View account details
5. Update contact details
6. Exit
Enter your choice: 1
Enter account holder name: Nandini Kumari
Enter initial deposit amount: 5000
Enter email address: nandini428@gmail.com
```

```
Enter initial deposit amount: 5000
Enter email address: nandini428@gmail.com
Enter phone number:
Account created successfully with Account Number: 1001
```

```
--- Welcome to the Banking Application ---
1. Create a new account
2. Deposit money
3. Withdraw money
4. View account details
```