K.R. Mangalam University

School of Engineering & Technology

Fundamentals Of Java Programming

Assignment 1

Banking Application for Accounts Management

Submitted by:

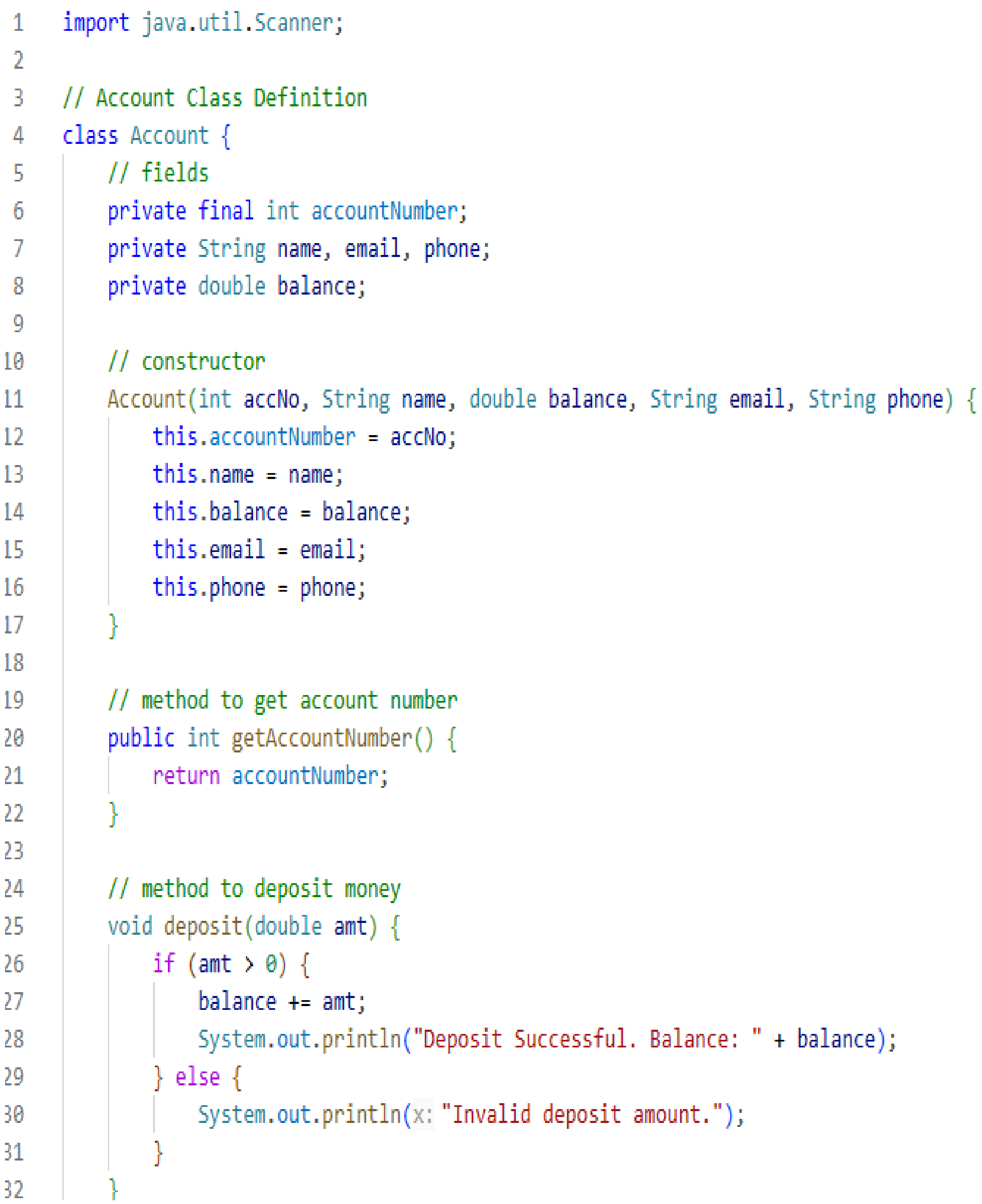
Name: Khushi

Roll No: 2401201102

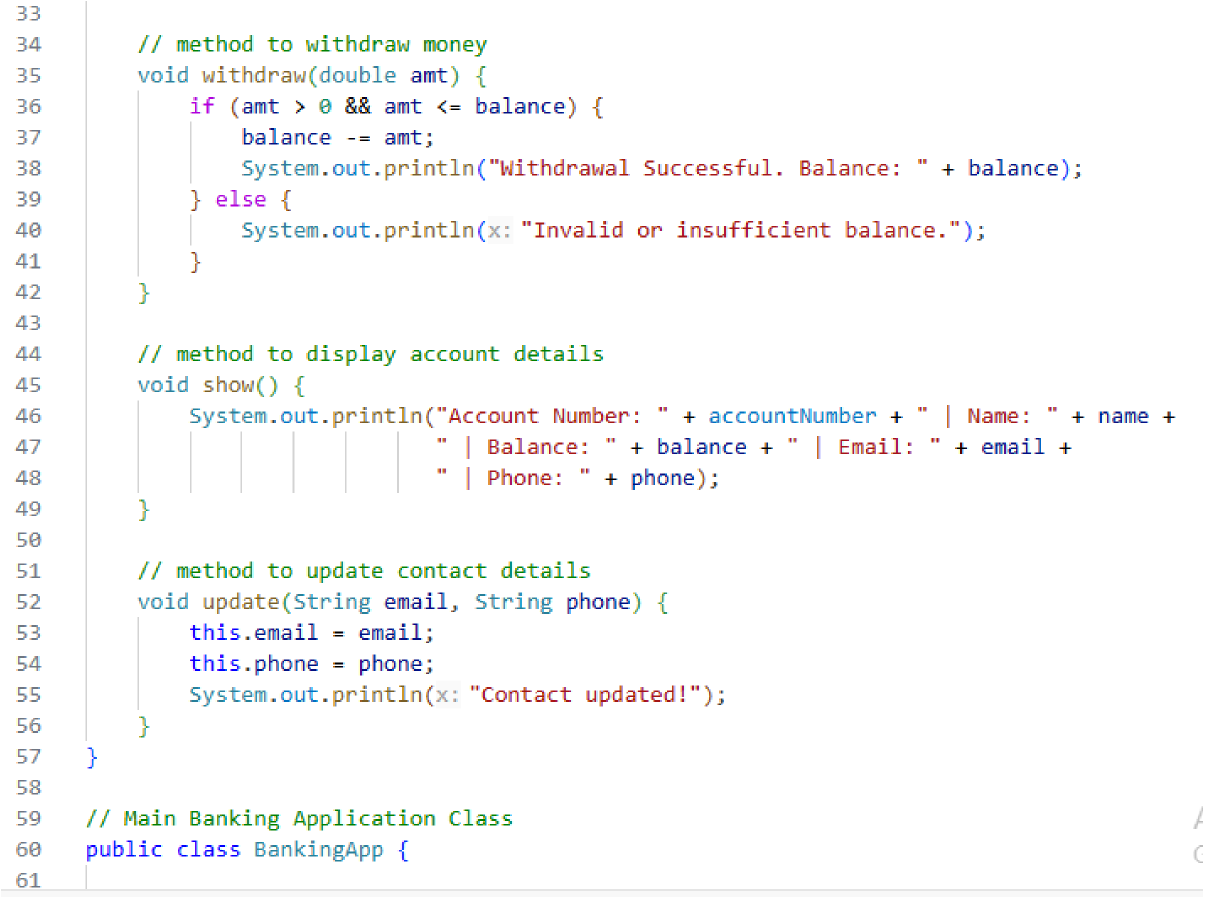
Course: BCA (AI & DS)

Section: B

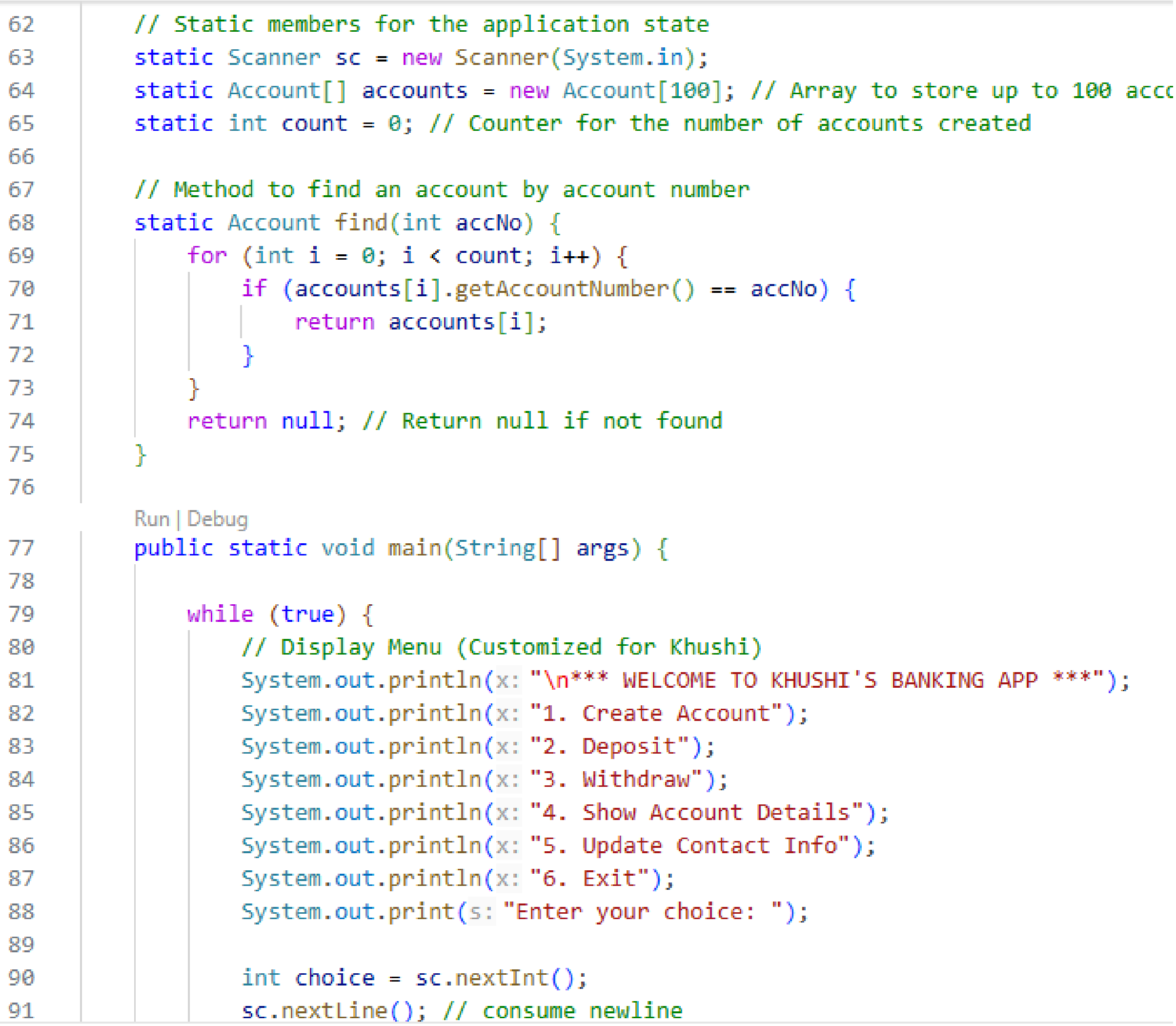
CODE:



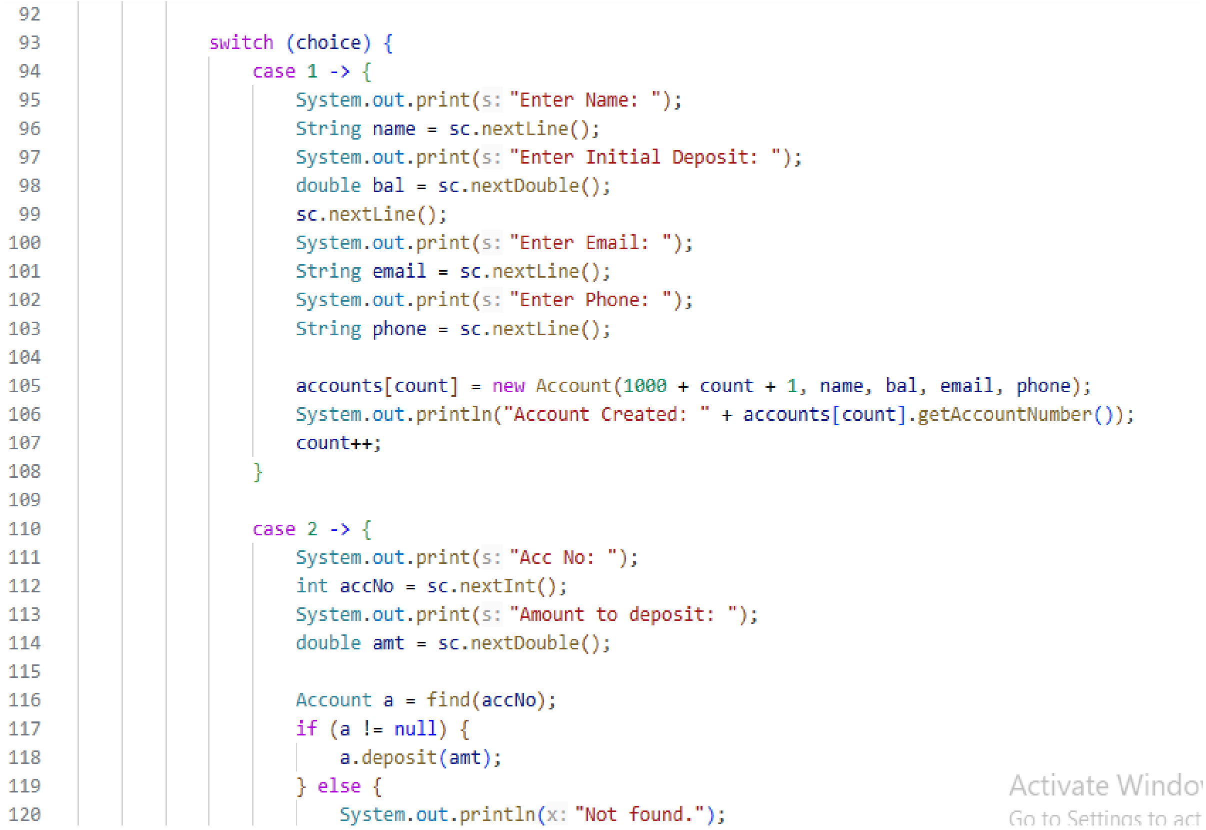
```java
import java.util.Scanner;

// Account Class Definition
class Account {
    // fields
    private final int accountNumber;
    private String name, email, phone;
    private double balance;

    // constructor
    Account(int accNo, String name, double balance, String email, String phone) {
        this.accountNumber = accNo;
        this.name = name;
        this.balance = balance;
        this.email = email;
        this.phone = phone;
    }

    // method to get account number
    public int getAccountNumber() {
        return accountNumber;
    }

    // method to deposit money
    void deposit(double amt) {
        if (amt > 0) {
            balance += amt;
            System.out.println("Deposit Successful. Balance: " + balance);
        } else {
            System.out.println(x: "Invalid deposit amount.");
        }
    }
```



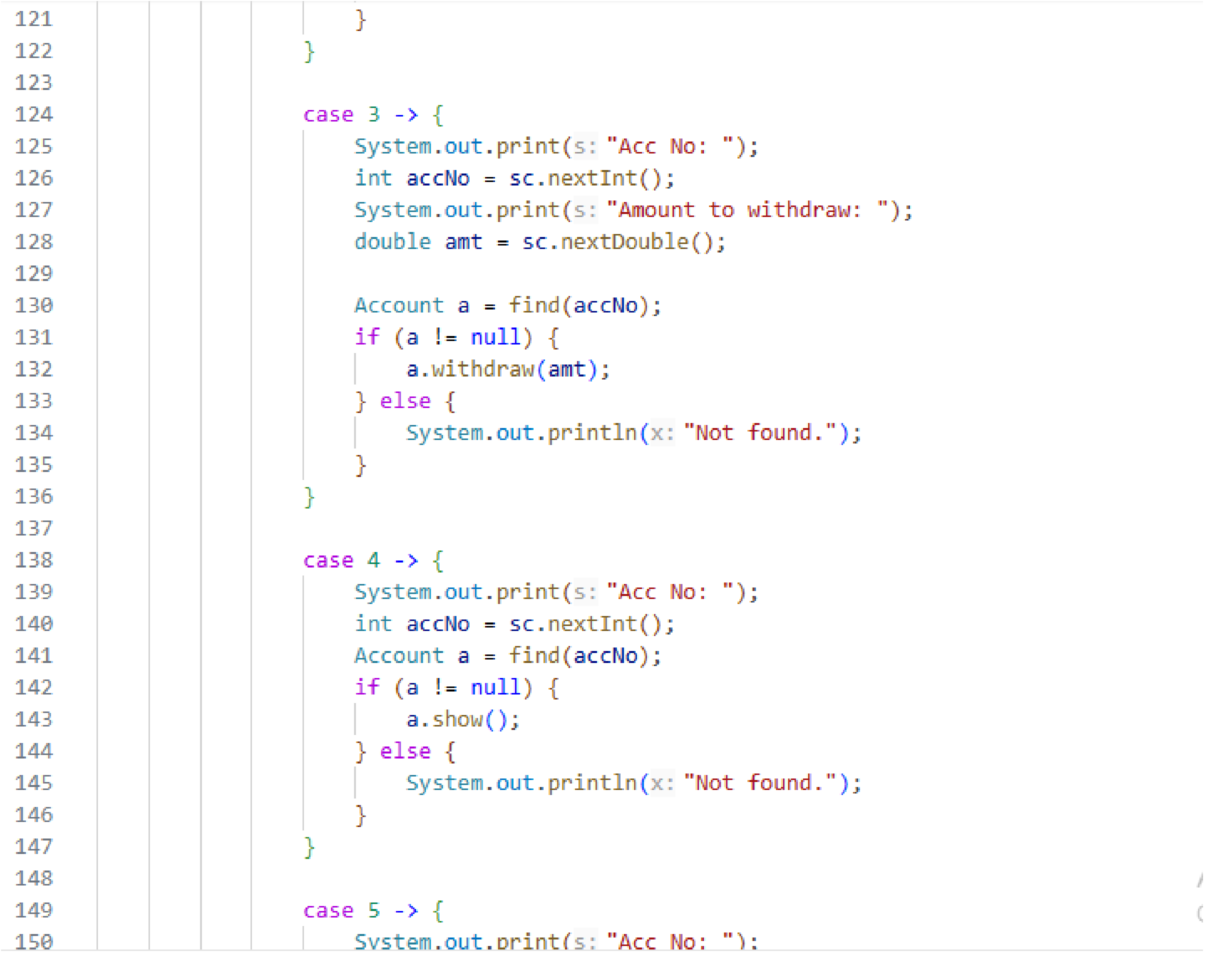
```java

    // method to withdraw money
    void withdraw(double amt) {
        if (amt > 0 && amt <= balance) {
            balance -= amt;
            System.out.println("Withdrawal Successful. Balance: " + balance);
        } else {
            System.out.println(x: "Invalid or insufficient balance.");
        }
    }

    // method to display account details
    void show() {
        System.out.println("Account Number: " + accountNumber + " | Name: " + name +
                                   " | Balance: " + balance + " | Email: " + email +
                                   " | Phone: " + phone);
    }

    // method to update contact details
    void update(String email, String phone) {
        this.email = email;
        this.phone = phone;
        System.out.println(x: "Contact updated!");
    }
}

// Main Banking Application Class
public class BankingApp {

```



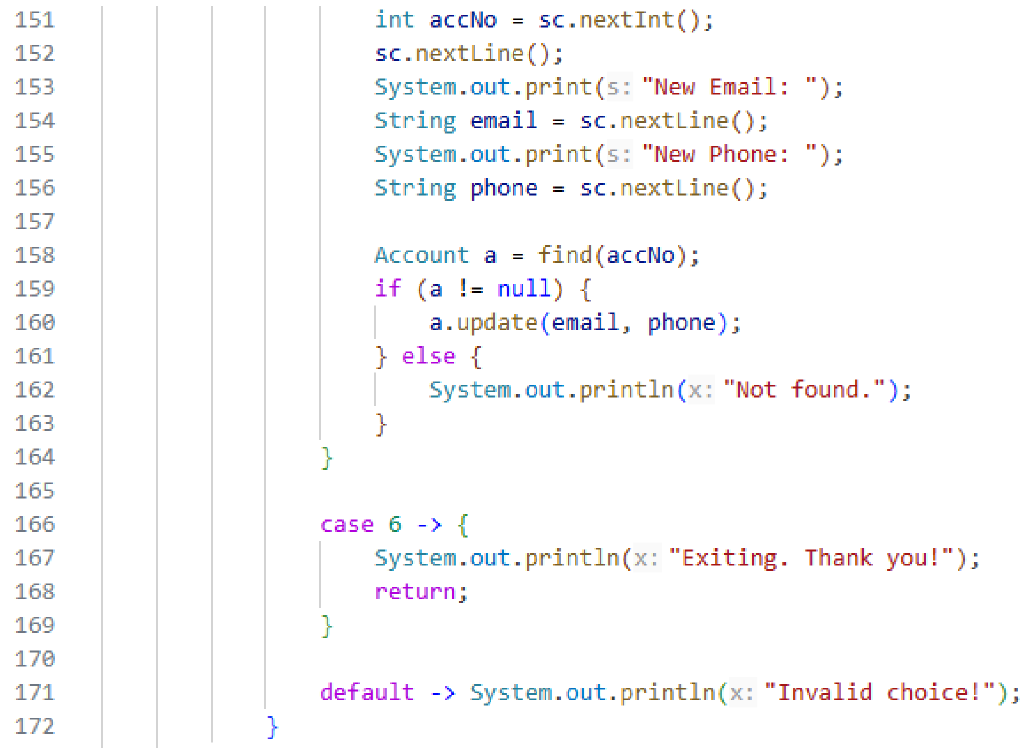
```java
    // Static members for the application state
    static Scanner sc = new Scanner(System.in);
    static Account[] accounts = new Account[100]; // Array to store up to 100 acc
    static int count = 0; // Counter for the number of accounts created

    // Method to find an account by account number
    static Account find(int accNo) {
        for (int i = 0; i < count; i++) {
            if (accounts[i].getAccountNumber() == accNo) {
                return accounts[i];
            }
        }
        return null; // Return null if not found
    }

    public static void main(String[] args) {

        while (true) {
            // Display Menu (Customized for Khushi)
            System.out.println(x: "\n*** WELCOME TO KHUSHI'S BANKING APP ***");
            System.out.println(x: "1. Create Account");
            System.out.println(x: "2. Deposit");
            System.out.println(x: "3. Withdraw");
            System.out.println(x: "4. Show Account Details");
            System.out.println(x: "5. Update Contact Info");
            System.out.println(x: "6. Exit");
            System.out.print(s: "Enter your choice: ");

            int choice = sc.nextInt();
            sc.nextLine(); // consume newline
```



```java

            switch (choice) {
                case 1 -> {
                    System.out.print(s: "Enter Name: ");
                    String name = sc.nextLine();
                    System.out.print(s: "Enter Initial Deposit: ");
                    double bal = sc.nextDouble();
                    sc.nextLine();
                    System.out.print(s: "Enter Email: ");
                    String email = sc.nextLine();
                    System.out.print(s: "Enter Phone: ");
                    String phone = sc.nextLine();

                    accounts[count] = new Account(1000 + count + 1, name, bal, email, phone);
                    System.out.println("Account Created: " + accounts[count].getAccountNumber());
                    count++;
                }

                case 2 -> {
                    System.out.print(s: "Acc No: ");
                    int accNo = sc.nextInt();
                    System.out.print(s: "Amount to deposit: ");
                    double amt = sc.nextDouble();

                    Account a = find(accNo);
                    if (a != null) {
                        a.deposit(amt);
                    } else {
                        System.out.println(x: "Not found.");
```

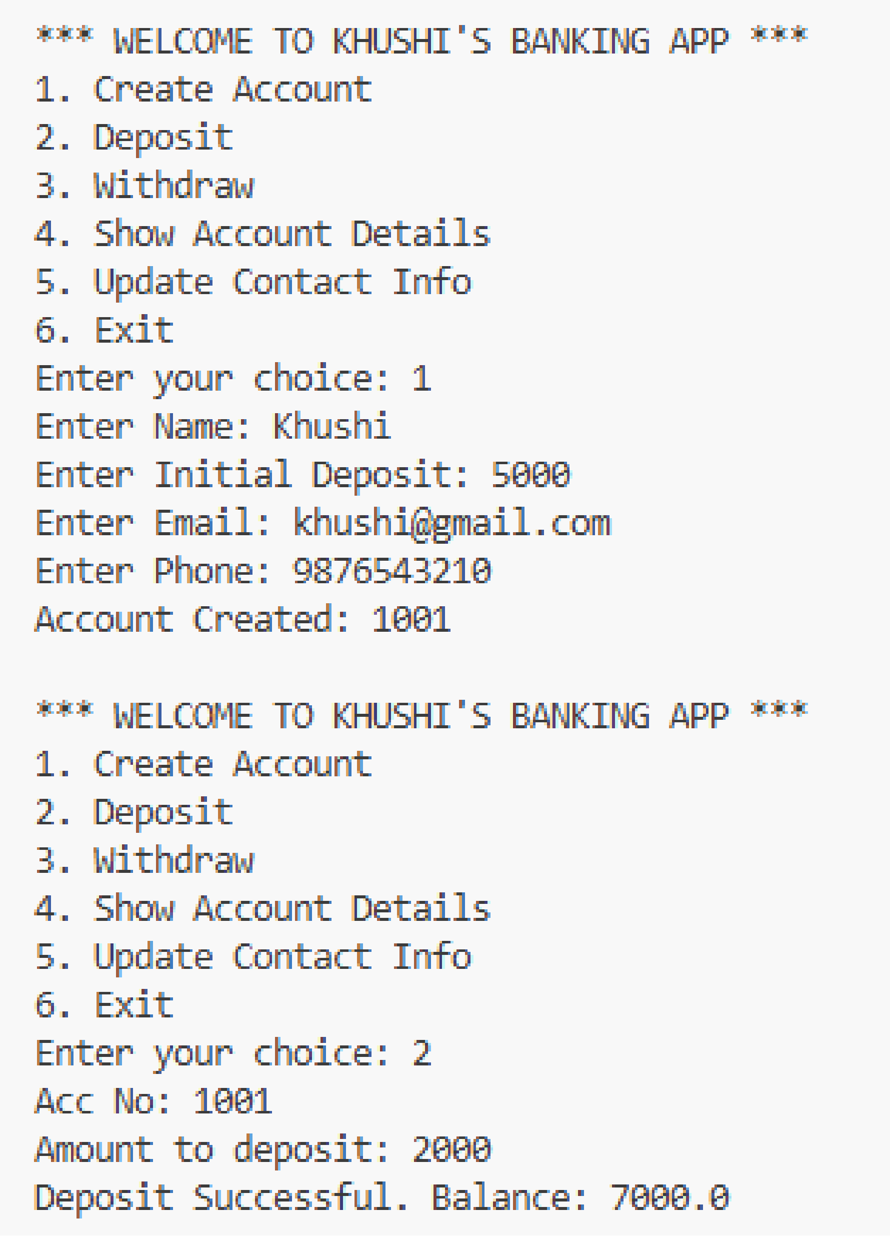


```
                    }
                }

                case 3 -> {
                    System.out.print(s: "Acc No: ");
                    int accNo = sc.nextInt();
                    System.out.print(s: "Amount to withdraw: ");
                    double amt = sc.nextDouble();

                    Account a = find(accNo);
                    if (a != null) {
                        a.withdraw(amt);
                    } else {
                        System.out.println(x: "Not found.");
                    }
                }

                case 4 -> {
                    System.out.print(s: "Acc No: ");
                    int accNo = sc.nextInt();
                    Account a = find(accNo);
                    if (a != null) {
                        a.show();
                    } else {
                        System.out.println(x: "Not found.");
                    }
                }

                case 5 -> {
                    System.out.print(s: "Acc No: ");
```

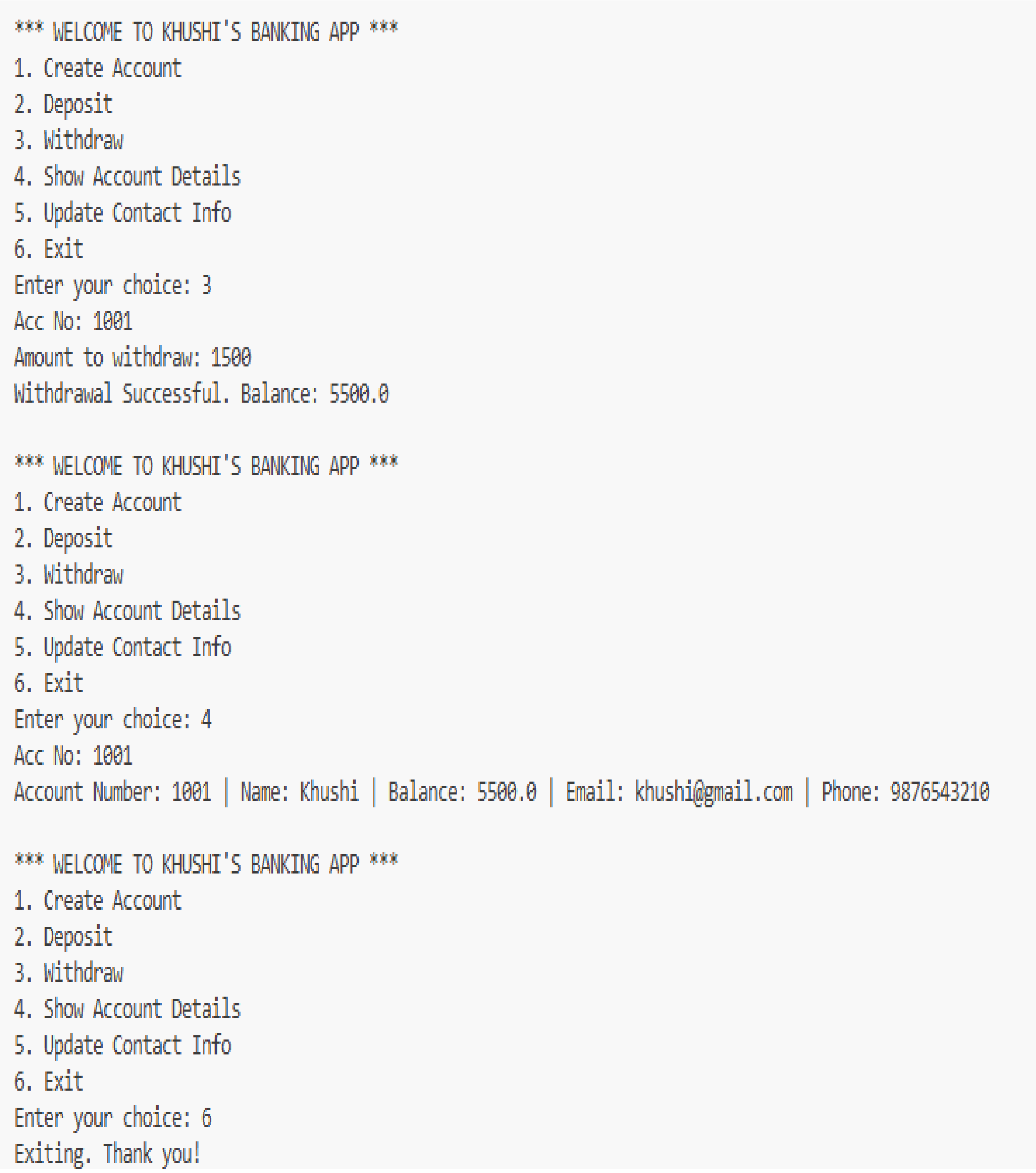


```
                        int accNo = sc.nextInt();
                        sc.nextLine();
                        System.out.print(s: "New Email: ");
                        String email = sc.nextLine();
                        System.out.print(s: "New Phone: ");
                        String phone = sc.nextLine();

                        Account a = find(accNo);
                        if (a != null) {
                            a.update(email, phone);
                        } else {
                            System.out.println(x: "Not found.");
                        }
                    }

                    case 6 -> {
                        System.out.println(x: "Exiting. Thank you!");
                        return;
                    }

                    default -> System.out.println(x: "Invalid choice!");
                }
```

OUTPUT:



```
*** WELCOME TO KHUSHI'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: Khushi
Enter Initial Deposit: 5000
Enter Email: khushi@gmail.com
Enter Phone: 9876543210
Account Created: 1001

*** WELCOME TO KHUSHI'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
```



```
*** WELCOME TO KHUSHI'S BANKING APP ***
1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit
Enter your choice: 3
Acc No: 1001
Amount to withdraw: 1500
Withdrawal Successful. Balance: 5500.0

*** WELCOME TO KHUSHI'S BANKING APP ***
1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit
Enter your choice: 4
Acc No: 1001
Account Number: 1001 | Name: Khushi | Balance: 5500.0 | Email: khushi@gmail.com | Phone: 9876543210

*** WELCOME TO KHUSHI'S BANKING APP ***
1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit
Enter your choice: 6
Exiting. Thank you!
```