Yash Kumar Gautam
2401730204, Section, B · AI/ML sem III

Assignment -1

(Q1)

```
import java.util.Scanner;
class Account {
    private int accountNumber;
    private string accountHolderName;
    private double balance;
    private string email;
    private string phoneNumber;
public Account(int accountNumber, string accountHolder
        Name, double initialDeposit, string email,
        string phoneNumber) {
    this.accountNumber = accountNumber;
    this.accountHolderName = accountHolderName;
    this.balance = initialDeposit;
    this.email = email;
    this.phoneNumber = phoneNumber;
}
public int getAccountNumber() {
    return accountNumber;
} public void deposit(double amount) {
    if (amount > 0) {
        balance += amount;
System.out.println("Deposit successful. New balance: " + balance);
} else
System.out.println("Invalid deposit amount. Amount must be
        positive.");
}
}
```

```
public void withdraw (double amount) {
    if (amount <= 0) {
    System.out.println ("Invalid withdrawal amount");
} else if (amount > balance) {
    system.out.println ("Insufficient balance")
} else {
    balance = amount;
system.out.println ("withdraw success);
}
}

public void displayAccountDetails() {
System.out.println ("account number: " + AccountNumber);
 system.out.println ("Balance: " + balance);
 System.out.println ("email: " + email);
 System.out.println ("phone number: " + phoneNumber);
}

public void updateContactDetails (String email, String
    phoneNumber)
        This.email = email;
        This.phonenumber = phoneNumber)
System.out.println ("contact details updates successfully");
}
}

public class UserInterface {
    private Account [] accounts;
    private int accountCount;
    private Scanner scanner;
    private int nextAccountNumber;
public UserInterface () {
    accounts = new accounts [100];
    accountCount = 0;
    Scanner = new Scanner (System.in);
```

```
Next    Account Number = 100;
}
    public void createAccount() {
        System.out.println("Enter account holder name:");
        String name = scanner.nextLine();
        double initialDeposit = 0;
        while (true) {
            System.out.println("Enter initial deposit amount:");
            String depositInput = scanner.nextLine();
            try {
                initialDeposit = Double.parseDouble(depositInput);
                if (initialDeposit < 0) {
                System.out.println("Initial deposit cannot be negative");
                } else {
                break;
            }
        } catch (NumberFormatException) {
            System.out.println("Invalid amount, please enter valid");
            }
        }
        System.out.println("Enter email address:");
        String email = scanner.nextLine();
        System.out.println("Enter phone number:");
        String phone = scanner.nextLine();
Account newAccount = new Account(NextAccountNumber, name,
                initialDeposit, email, phone);
accounts[accountCount] = newAccount;
    accountCount++;
System.out.println("Account created successfully with AccountNumber:
                newAccountNumber);
        NextAccountNumber++;
    }
```

```
private Account findAccountByNumber(int accountNumber) {
    for (int i = 0; i < accountCount; i++) {
        if (accounts[i].getAccountNumber() == accountNumber) {
            return accounts[i];
        }
    }
    return null;
}
public void performDeposit() {
    System.out.println("Enter account number:");
    int accNum = readIntInput();
    Account acc = findAccountByNumber(accNum);
    if (acc == null) {
        System.out.println("Account not found");
        return;
    }
    System.out.println("Enter amount to deposit:");
    double amount = readDoubleInput();
    acc.deposit(amount);
}
public void performWithdrawal() {
    System.out.println("Enter account number:");
    int accNum = readIntInput();
    Account acc = findAccountByNumber(accNum);
    if (acc == null) {
        System.out.println("account not found");
        return;
    }
System.out.println("Enter amount to withdraw");
    double amount = readDoubleInput();
    acc.withdraw(amount);
}
```

```
public void showAccountDetails() {
    System.out.println("Enter account number:");
    int accNum = readIntInput();
    Account acc = findAccountByNumber(accNum);
    if (acc == null) {
        System.out.println("account not found");
        return;
    }
    acc.displayAccountDetails();
}
public void updateContact() {
    System.out.println("enter account number:");
    int accNum = readIntInput();
    Account acc = findAccountByNumber(accNum);
    if (acc == null) {
        System.out.println("account not found");
        return;
    }
    System.out.println("Enter new email address:");
    String email = scanner.nextLine();
    System.out.println("Enter new phone number");
    String phone = scanner.nextLine();
    acc.updateContactDetails(email, phone);
}
public void mainMenu() {
    while (true) {
        System.out.println("\nWelcome to the Banking Application");
        System.out.println("1. Create a new account");
        System.out.println("2. Deposit money");
```

```
System.out.println("4 View account details");
system.out.println("5 Update contact details");
system.out.println("6. Exit");
system.out.println("Enter your choice");

int choice = readIntInput();
    switch(choice) {
    case 1:
        createAccount();
        break;
case 2:
    performDeposit();
     break;
Case 3;
    performwithdrawal();
     break;
case 4;
    ShowAccountDetails();
    break;
case 5:
    updateContact();
    break;
    case 6;
system.out.println("Thank you for using the application");
    scanner.close();
     return;
     default;
system.out.println("Invalid choice please enter a
                    number between 1 and 6");
```

```
private int readInt Input() {
        while (true) {
    String input = scanner.nextLine();
        try {
    initial = integer.parseInt(input);
        return void;
} catch (number format Exception) {
    System.out.println("Invalid Input. Reenter");
}
}
}

private double readDouble Input() {
        while (true) {
    String input = scanner.nextLine();
        try {
        double val = Double.parseDouble(input);
            return val;
} catch (number format Exception e) {
System.out.println("Invalid Input please enter a
                valid number: ");
}
}

public static void main (String[] args) {
        User Interface ui = new user interface();
        ui.mainmenu();
}
```