Name : Swastik Sharma
Roll no. : 2401730032
Section - B

JAVA ASSIGNMENT - 1

```
import java.util.scanner;

// This is the account class
class Account {
    private int accountNumber;
    private string accountHolder name;
    private double balance;
    private string email;
    private string phone number;
// Constructor
  public Account (int accountNumber, string accountHolderName
                  , double balance, string email, string phone
                    number) {
        this.accountNumber = accountNumber;
        this.accountHolderName = accountHolderName;
        this.balance = balance;
        this.email = email;
        this.phonenumber = phoneNumber;
}
// method to deposit money.
  public void deposit(double amount){
        if (amount > 0) {
          balance += amount
        System.out.println("Amount in rupees" + "amount" +
                           "deposited" successfully! New balance:"
                                + balance);
    } else {
      System.out.println("Invalid amount. Deposit failed")
}
```

```
// method to withdraw money
public void withdraw (double amount) {
    if (amount > 0 && amount <= balance) {
        balance -= amount;
        System.out.println("Amount withdrawn successfully
            new balance: " + balance);
    } else {
        system.out.println("Invalid amount or unsufficient
            balance");
    }
}

// method to access private member
public int getAccountNumber () {
    return accountNumber;
}

// method to update contact details
public void updateContactDetails (String email, String
    phone number) {
    this.email = email;
    this.phoneNumber = phone number;
    system.out.println("Contact details updated successfully");
}
}

// This user interface closes which has been kept same as file
name, i.e, Main.
public class Main {
    private Account [] accounts;
    private int accountCount;
    private scanner sc;
public Main (int size) {
    account = new account [size];
    accountCount = 0;
```

```
Sc = new Scanner (System.in);
}
// This is the method to create new account, that takes
account details from & stores in the array.
    public void createAccount () {
        System.out.println ("Enter Account holder name:");
        String name = sc.nextLine ();
        System.out.println ("Enter initial deposit amount:");
        double amount = sc.nextDouble ();
        Sc.nextLine();
        Sout ("Enter email:");
        String email = sc.next line ();
        Sout ("Enter phone number:")
        string phone = sc.nextline ();
        int account number = 1001 + account count;
        account [account count] = new Account (account Number,
        name, amount, email, phone);
        account count ++;
        Sout ("Account created successfully with Account
                Number " + account Number);
    }
// This is method to handle deposit operation
    public void performDeposit () {
        Sout = ("Enter acc. Number: ");
        int acc No. = sc.next line ();
        Sout ("Enter amt. to deposit:");
        double amount = sc.next double ();
        sc next lin ();
Account acc = find Account (acc No);
    if (acc != null) {
```

```
        acc.deposit (amount);
    } else {
        Sout ("Account not found ");
    }
}
// This method to perform withdraw operation
    public void perform withdrawal () {
    Sout ("Enter Acc.Number : ");
     int accNo . sc next Int ();
      sc. next line;
Account acc = find Account (acc No);
  if (acc ! = null) {
    Sout ("Enter amt . to withdraw:");
      Sc. next line ();
    acc. withdraw (amount );
} else {
  Sout ("Account not found ! ");
}
}
// This method to show Account details
  Public void show Account Details () {
      Sout ("Enter Account Number : ");
    int accNo . = sc . next Int ();
    SC . next line ();
Account acc = find Account (acc No);
    if (acc ! = null);
    acc . display Account Details ();
} else {
  Sout ("Account not found : ");
}
```

```
// find Account holder
private Account find Account (int accNo) {
    for (int i=0 ; i < amount count ; i++) {
        if (accounts [i] get Account Number (I= accNo)
{
    return accounts [i];
    }
}
return null;
}
// Main menu that handle user choice
    public void main menu () {
     int choice ;
   do {
   Sout ("Welcome to the Banking Application!");
    Sout ("1. Create new Account");
    Sout ("2. Deposit money");
    sout ("3. Withdraw money");
    Sout ("4. View Account detail");
    sout ("5. Update contact Detail");
    sout ("6. Exit");
    Sout ("Enter your choice");
    choice .sc. next int ();
       ~~Sc next line~~
       Sc. next line ();
  Switch (choice) {
Case 1: Create Account (); break;
case 2: Perform Deposit (); break;
case 3: perform Withdrawal (); brea;
```

```
Case 4: showAccountDetails(); break;
Case 5: updateContact(); break;
Case 6: sout("Thank you"); break;
default: sout("Invalid choice");
}
} while(choice != 6);
}
// Main Method.
public static void main(String[] args) {
    Main ui = new Main(100); // Max 100 accounts
    ui.mainMenu();
}
}
```

SWASTIK SHARMA
2401730032
Sec-B (Btech CSE AI/ML)