

```

import java.util.Scanner;

abstract class Account {
    String customerName;
    String accountNumber;
    double balance;
    String accountType;

    public Account(String customerName, String accountNumber, String accountType, double balance)
    {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    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.");
        }
    }

    public void displayBalance() {
        System.out.println("Account Balance: " + balance);
    }

    public abstract void withdraw(double amount);

    public abstract void updateBalance();
}

class CurAcct extends Account {
    private static final double MINIMUM_BALANCE = 1000;
    private static final double SERVICE_CHARGE = 50;

    public CurAcct(String customerName, String accountNumber, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    public void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
            updateBalance();
        } else {

```

```

        System.out.println("Insufficient balance.");
    }
}

public void updateBalance() {
    if (balance < MINIMUM_BALANCE) {
        balance -= SERVICE_CHARGE;
        System.out.println("Service charge imposed due to low balance. New balance: " + balance);
    }
}
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.05;

    public SavAcct(String customerName, String accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }

    public void computeInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest computed and added. New balance: " + balance);
    }

    public void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    public void updateBalance() {
        computeInterest();
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter Customer Name: ");
        String name = scanner.nextLine();

        System.out.print("Enter Account Number: ");
        String accNum = scanner.nextLine();
    }
}

```

```

System.out.print("Enter Account Type (Savings/Current): ");
String type = scanner.nextLine();

System.out.print("Enter Initial Balance: ");
double balance = scanner.nextDouble();

Account account;
if (type.equalsIgnoreCase("Savings")) {
    account = new SavAcct(name, accNum, balance);
} else if (type.equalsIgnoreCase("Current")) {
    account = new CurAcct(name, accNum, balance);
} else {
    System.out.println("Invalid Account Type.");
    scanner.close();
    return;
}

boolean quit = false;
while (!quit) {
    System.out.println("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4. Update Balance\n5.
Quit");
    System.out.print("Enter your choice: ");
    int choice = scanner.nextInt();

    switch (choice) {
        case 1:
            System.out.print("Enter amount to deposit: ");
            double depositAmount = scanner.nextDouble();
            account.deposit(depositAmount);
            break;
        case 2:
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            account.withdraw(withdrawAmount);
            break;
        case 3:
            account.displayBalance();
            break;
        case 4:
            account.updateBalance();
            break;
        case 5:
            quit = true;
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
    }
}

```

```
    }  
    System.out.println("Thank you for banking with us.");  
  }  
}
```

Enter Customer Name: Shonal
Enter Account Number: 12234
Enter Account Type (Savings/Current): Savings
Enter Initial Balance: 1000000

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance

5. Quit

Enter your choice: 1
Enter amount to deposit: 10000
Deposit successful. New balance: 1010000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance

5. Quit

Enter your choice: 3
Account Balance: 1010000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance

5. Quit

Enter your choice: 4
Interest computed and added. New balance: 1060500.0

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance

5. Quit

Enter your choice: 5
Thank you for banking with us.

Lab - Program - 5

- 8.) Develop a Java Program, to create a class Bank that maintains two kinds of ~~of~~ for its customers, one called savings account and the other current account. The savings account provides compound interest and with draft facilities but no cheque book facility. The current account provides cheque book facilities but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a penalty charge is imposed. Create a class account that stores customers name, account no. and type of account. From this derive the classes..

```
import java.util.Scanner;  
class Account {
```

```
    String customer Name, account Number,  
    account Type;  
    double balance;  
    public Account (String name, String  
        acc No, String type, double bal) {  
        customer Name = name;  
        account Number = acc No;
```



```

AccountType = type;
balance = bal;
}
public void deposit (double amt) {
    if (amt < 0) {
        System.out.println("Deposit amount must be positive:");
        return;
    }
    balance += amt;
    System.out.println("Deposit successful. updated balance: " + balance);
}
public void displayBalance() {
    System.out.println("Current Account Balance: " + balance);
}
}
}

```

```

class SavAcct extends Account {
    static final double interestRate = 0.05;
    public SavAcct (String name, String accNo, double bal) {
        super (name, accNo, "Savings", bal);
    }
    public void calcInterest (int years) {
        double interest = balance * Math.pow(1 + interestRate, years) - balance;
        balance += interest;
    }
}

```



```

System.out.println("Interest of " + interest + "
deposited. Updated balance: " + balance);
}

```

```

public void withdraw (double amt) {
    if (amt <= balance) {
        balance -= amt;
        System.out.println("Withdrawal
        Successful. Updated balance: " + balance);
    }
}

```

```

else {
    System.out.println("Insufficient
    balance.");
}
}

```

```

class CurrentAccount extends Account {

```

```

    static final double minBalance = 10000;

```

```

    static final double penaltyCharge = 50;

```

```

    public CurrentAccount (String name, String accNo,
    double bal) {

```

```

        super (name, accNo, "Current", bal);
    }

```

```

    public void withdraw (double amount) {

```

```

        if (amt <= balance) {

```

```

            balance -= amount;

```

```

            System.out.println("Withdrawal successful
            Updated balance: " + balance);

```

```

        }
        if (balance < minBalance) {

```

```

            balance -= penaltyCharge;

```


System.out.println("Balance below minimum
Service charge of " + penalty charge +
" imposed.");

System.out.println("Updated
balance: " + balance);

}

else {

System.out.println("Insufficient
balance");

}

}

3.

public class Bank {

public static void main(String[] args) {

Scanner SC = new Scanner(System.in);

SavAcct SA = new SavAcct("ABC",
"SA123", 1000);

CurAcct CA = new CurAcct("XYZ",
"CA456", 600);

SA.deposit(200);

SA.calculateInterest(2);

SA.withdraw(500);

SA.displayBalance();

CA. deposit (300);
CA. withdrawal (700);
CA. display Balance ();

}

}

Output:-

Deposit Successful. Updated Balance: 1200.0
Interest of 123.0 deposited. Updated
Balance: 1323.0

Withdrawal Successful. Updated Balance: 823.0
Current Account Balance: 823.0.

Deposit Successful. Updated Balance: 900.0
Withdrawal Successful. Updated Balance: 823.0
Balance below minimum. Service -