**Name:** Makwana Dainik Kalabhai

**Roll no.:** 3133

**Division:** B

**Subject:** JAVA Practical

**Assignment-2**

**Question-1**

package com.cafe;

import java.util.ArrayList;

import java.util.Scanner;

interface CoffeeOrder {

    void selectCoffee(String coffeeType);

    void selectSize(String size);

    void displayOrder();

    int getPrice();

}

class CafeOrder implements CoffeeOrder {

    String coffeeType;

    double price;

    ArrayList<ArrayList<Object>> Orders = new ArrayList<ArrayList<Object>>();

    public void selectCoffee(String coffeeType) {

        this.coffeeType = coffeeType;

        CafeOrder C1 = new CafeOrder();

        if(coffeeType == "Latte")

            this.price = 160;

        if(coffeeType == "Cappuccino")

            this.price = 260;

        if(coffeeType == "Caffè mocha")

            this.price = 200;

        if(coffeeType == "Espresso")

            this.price = 210;

        if(coffeeType == "Frappe")

            this.price = 250;

        if(coffeeType == "Americano")

            this.price = 170;

    }

    public void selectSize(String size) {

        ArrayList<Object> ind = new ArrayList<Object>();

        ind.add(this.coffeeType);

        ind.add(size);

        if(size=="small")

            ind.add(this.price/2);

        if(size=="medium")

            ind.add(this.price/1.5);

        if(size=="large")

            ind.add(this.price);

        Orders.add(ind);

    }

    public void displayOrder() {

        double total = 0;

        System.out.print("\n\nBill Details...\n------------------------------------");

        for(int i=0; i<Orders.size(); i++)

        {

            System.out.printf("\n%d) %s (%s): Rs.%.2f",i+1,Orders.get(i).get(0),Orders.get(i).get(1),Orders.get(i).get(2));

            total += (double)Orders.get(i).get(2);

        }

        System.out.println("\n\nTotal Payable Amount: Rs."+total+"\n");

    }

    public int getPrice() {

        return 0;

    }

}

class Que1 {

    static CafeOrder C = new CafeOrder();

    static Scanner S = new Scanner(System.in);

    static void chooseType()

    {

        while(true)

        {

            System.out.println("\nCoffee Types...");

            System.out.println("  1) Latte (Rs.160)");

            System.out.println("  2) Cappuccino (Rs.260)");

            System.out.println("  3) Caffè mocha (Rs.200)");

            System.out.println("  4) Espresso (Rs.210)");

            System.out.println("  5) Frappe (Rs.250)");

            System.out.println("  6) Americano (Rs.170)");

            System.out.println("  7) Exit");

            System.out.print("Choose Coffee Type: ");

            int choice = S.nextInt();

            switch(choice)

            {

                case 1:

                    C.selectCoffee("Latte");

                    chooseSize();

                    break;

                case 2:

                    C.selectCoffee("Cappuccino");

                    chooseSize();

                    break;

                case 3:

                    C.selectCoffee("Caffè mocha");

                    chooseSize();

                    break;

                case 4:

                    C.selectCoffee("Espresso");

                    chooseSize();

                    break;

                case 5:

                    C.selectCoffee("Frappe");

                    chooseSize();

                    break;

                case 6:

                    C.selectCoffee("Americano");

                    chooseSize();

                    break;

                default:

                    return;

            }

        }

    }

    static void chooseSize()

    {

        System.out.println("\nCoffee Sizes(By Default Large)...");

        System.out.printf("  1) Small (Rs.%.2f)",Math.ceil(C.price/2));

        System.out.printf("\n  2) Medium (Rs.%.2f)",Math.ceil(C.price/1.5));

        System.out.printf("\n  3) Large (Rs.%.2f)",C.price);

        System.out.print("\n\nChoose Coffee Size: ");

        int choice = S.nextInt();

        switch(choice)

        {

            case 1:

                C.selectSize("small");

                break;

            case 2:

                C.selectSize("medium");

                break;

            default:

                C.selectSize("large");

                break;

        }

    }

    public static void main(String args[])

    {

        while(true)

        {

            System.out.println("\nChoices...");

            System.out.println("  1) Place an Order");

            System.out.println("  2) Display Bill");

            System.out.println("  3) Exit");

            System.out.print("Enter your Choice: ");

            int choice = S.nextInt();

            switch(choice)

            {

                case 1:

                    chooseType();

                    break;

                case 2:

                    try {

                        C.Orders.get(0);

                        C.displayOrder();

                        return;

                    }

                    catch(Exception e) {

                        System.out.println("\*\*\*Please! Place Order First\*\*\*");

                    }

                    break;

                default:

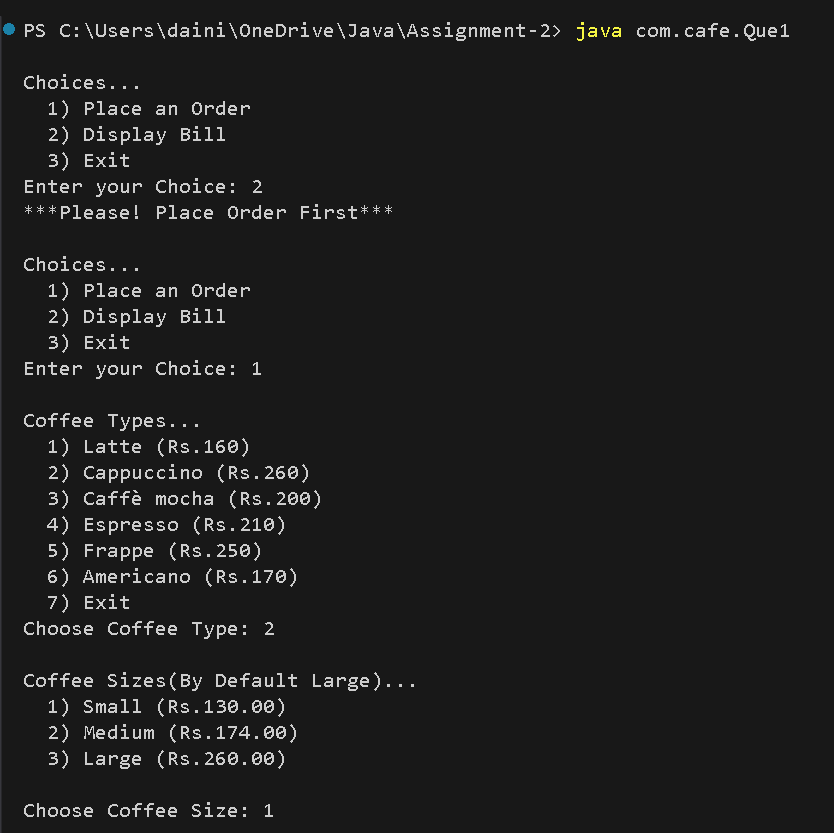
                    return;

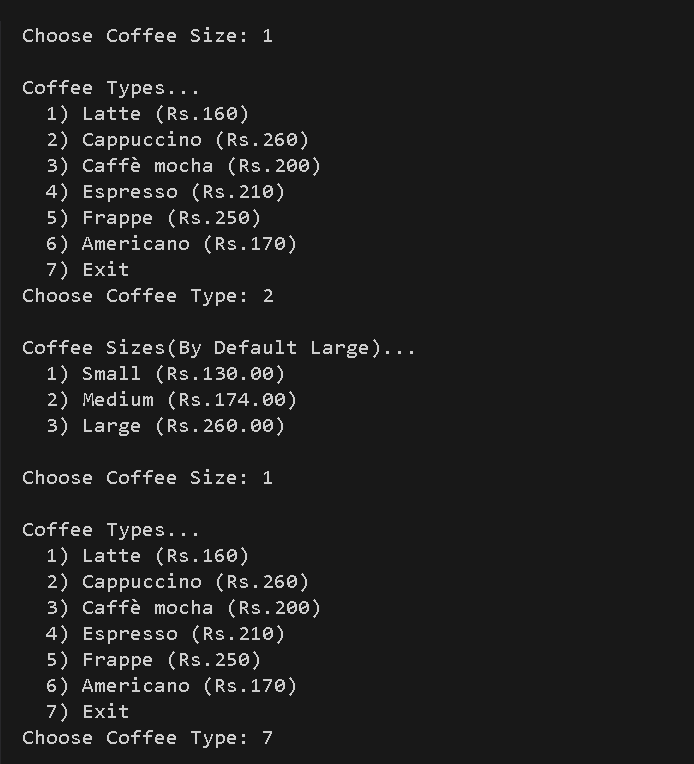
            }

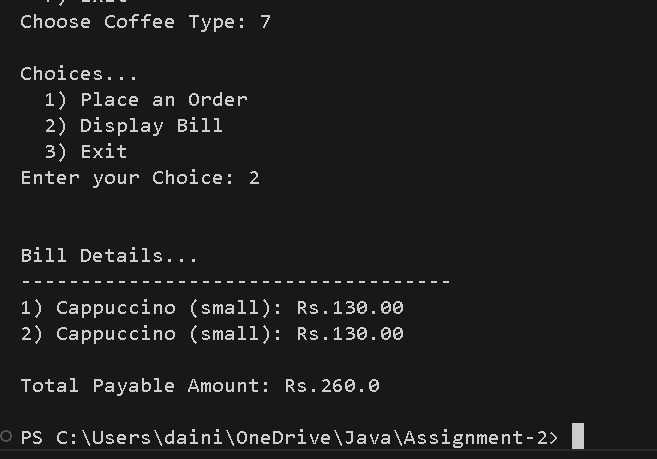
        }

    }

}







**Question-2**

package com.kiosk;

import java.util.Scanner;

interface DenominationHandler

{

    void FiveHundredHandler(long amount);

    void OneHundredHandler(long amount);

    void TenHandler(long amount);

    void FiveHandler(long amount);

}

class Kiosk implements DenominationHandler {

    static Kiosk K = new Kiosk();

    int fiveHundred = 0;

    int oneHundred = 0;

    int ten = 0;

    int five = 0;

    static long custNo;

    static long payAmount;

    int rem;

    Kiosk() {

    }

    Kiosk(long custNo, long payAmount)

    {

        this.custNo = custNo;

        this.payAmount = payAmount;

        K.FiveHundredHandler(payAmount);

    }

    public void FiveHundredHandler(long amount) {

        while(amount>=500)

        {

            amount -= 500;

            fiveHundred++;

        }

        K.OneHundredHandler(amount);

    }

    public void OneHundredHandler(long amount) {

        while(amount>=100)

        {

            amount -= 100;

            oneHundred++;

        }

        K.TenHandler(amount);

    }

    public void TenHandler(long amount) {

        while(amount>=10)

        {

            amount -= 10;

            ten++;

        }

        K.FiveHandler(amount);

    }

    public void FiveHandler(long amount) {

        while(amount>=5)

        {

            amount -= 5;

            five++;

        }

        this.rem = (int)amount;

        K.displayDetails();

    }

    public void displayDetails() {

        System.out.println("\n\nYour Bill Details...\n------------------------------------");

        System.out.println("Customer Number: "+custNo);

        System.out.printf("\n500 \* %d  :    Rs.%d",fiveHundred,fiveHundred\*500);

        System.out.printf("\n100 \* %d  :    Rs.%d",oneHundred,oneHundred\*100);

        System.out.printf("\n10 \* %d   :    Rs.%d",ten,ten\*10);

        System.out.printf("\n5 \* %d    :    Rs.%d",five,five\*5);

        System.out.println("\nRemaining Amount: Rs."+rem);

        System.out.println("\nTotal Payable Amount: Rs."+payAmount+"\n");

    }

}

class Que2 {

    public static void main(String[] args)

    {

        Scanner S = new Scanner(System.in);

        System.out.print("\nEnter Customer Number: ");

        long custNo = S.nextLong();

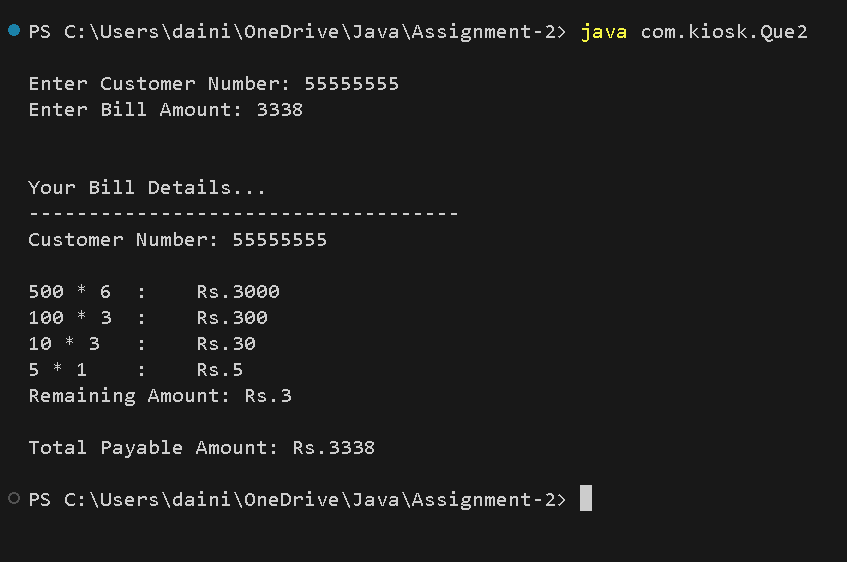
        System.out.print("Enter Bill Amount: ");

        long amount = S.nextLong();

        Kiosk K1 = new Kiosk(custNo,amount);

    }

}



**Question-3**

package com.bank.exceptions;

public class BankingException extends Exception {

    public BankingException(String str) {

        super(str);

    }

}

package com.bank;

import java.util.Scanner;

import com.bank.exceptions.BankingException;

interface BankingOperations {

    void deposit(double amount) throws Exception;

    void withdraw(double amount) throws Exception;

    double getBalance();

}

class Account implements BankingOperations {

    private long accNo;

    private double balance;

    Account(long accNo, double balance) {

        this.accNo = accNo;

        this.balance = balance;

    }

    long getAccNo() {

        return this.accNo;

    }

    @Override

    public double getBalance() {

        return this.balance;

    }

    @Override

    public void deposit(double amount) throws Exception {

        if(amount > 0) {

            this.balance += amount;

            System.out.printf("\n\*\*\*Rs.%.2f Deposited Successfully\*\*\*\n",amount);

            System.out.println("Balance: Rs."+this.balance);

        }

        else {

            throw new BankingException("\*\*\*Deposit Amount is invalid\*\*\*");

        }

    }

    @Override

    public void withdraw(double amount) throws Exception {

        if(amount <= 0) {

            throw new BankingException("\*\*\*Withdrawal Amount must be greater than 0\*\*\*");

        }

        if(amount>this.balance) {

            throw new BankingException("\*\*\*Insufficient Balance\*\*\*");

        }

        else {

            this.balance -= amount;

            System.out.printf("\n\*\*\*Rs.%.2f Withdraw Successfully\*\*\*\n",amount);

            System.out.println("Balance: Rs."+this.balance);

        }

    }

}

class BankingSystem extends Account {

    BankingSystem(long accNo, double balance) {

        super(accNo, balance);

        System.out.println("Account created Successfully");

        System.out.println("Account Number: "+this.getAccNo());

    }

    void retrieveInfo() {

        System.out.println("Account Number: "+this.getAccNo());

        System.out.println("Balance: Rs."+this.getBalance());

    }

}

public class Que3 {

    public static void main(String[] args) {

        Scanner S = new Scanner(System.in);

        BankingSystem B[] = new BankingSystem[15];

        int index = 0;

        long accNo;

        double amount;

        boolean checkAcc = false;

        while(true) {

            System.out.println("\nChoices...");

            System.out.println("1) Create Account");

            System.out.println("2) Deposit Money");

            System.out.println("3) Withdraw Money");

            System.out.println("4) Retrieve Information");

            System.out.println("Other to Exit");

            System.out.print("\nEnter your choice: ");

            int choice = S.nextInt();

            switch(choice) {

                case 1:

                    if(index<15) {

                        accNo = (int)(Math.random()\*111111111);

                        for(int i=0; i<index; i++) {

                            if(B[i].getAccNo() == accNo) {

                                accNo = (int)(Math.random()\*111111111);

                            }

                        }

                        B[index] = new BankingSystem(accNo, 0);

                        index++;

                    }

                    else {

                        System.out.println("\*\*\*You exceeded the maximum limit of account creation. So, You can't Create the Account\*\*\*");

                    }

                    break;

                case 2:

                    System.out.print("\nEnter the Account Number: ");

                    accNo = S.nextLong();

                    for(int i=0; i<index; i++) {

                        if(B[i].getAccNo() == accNo) {

                            System.out.print("Enter the Amount which you want to deposit: ");

                            amount = S.nextDouble();

                            try {

                                B[i].deposit(amount);

                            }

                            catch(Exception e)

                            {

                                System.out.println("\n"+e.getMessage());

                            }

                            checkAcc = true;

                        }

                    }

                    if(!checkAcc) {

                        System.out.println("\*\*\*Acount Number: "+accNo+" is not exist\*\*\*");

                    }

                    checkAcc = false;

                    break;

                case 3:

                    System.out.print("\nEnter the Account Number: ");

                    accNo = S.nextLong();

                    for(int i=0; i<index; i++) {

                        if(B[i].getAccNo() == accNo && B[i].getBalance()>0) {

                            System.out.print("Enter the Amount which you want to withdraw: ");

                            amount = S.nextDouble();

                            try {

                                B[i].withdraw(amount);

                            }

                            catch(Exception e)

                            {

                                System.out.println("\n"+e.getMessage());

                            }

                            checkAcc = true;

                        }

                    }

                    if(!checkAcc) {

                        System.out.println("\*\*\*Acount Number: "+accNo+" is not exist\*\*\*");

                    }

                    checkAcc = false;

                    break;

                case 4:

                    System.out.print("\nEnter the Account Number: ");

                    accNo = S.nextLong();

                    for(int i=0; i<index; i++) {

                        if(B[i].getAccNo() == accNo) {

                            B[i].retrieveInfo();

                            checkAcc = true;

                        }

                    }

                    if(!checkAcc) {

                        System.out.println("\*\*\*Acount Number: "+accNo+" is not exist\*\*\*");

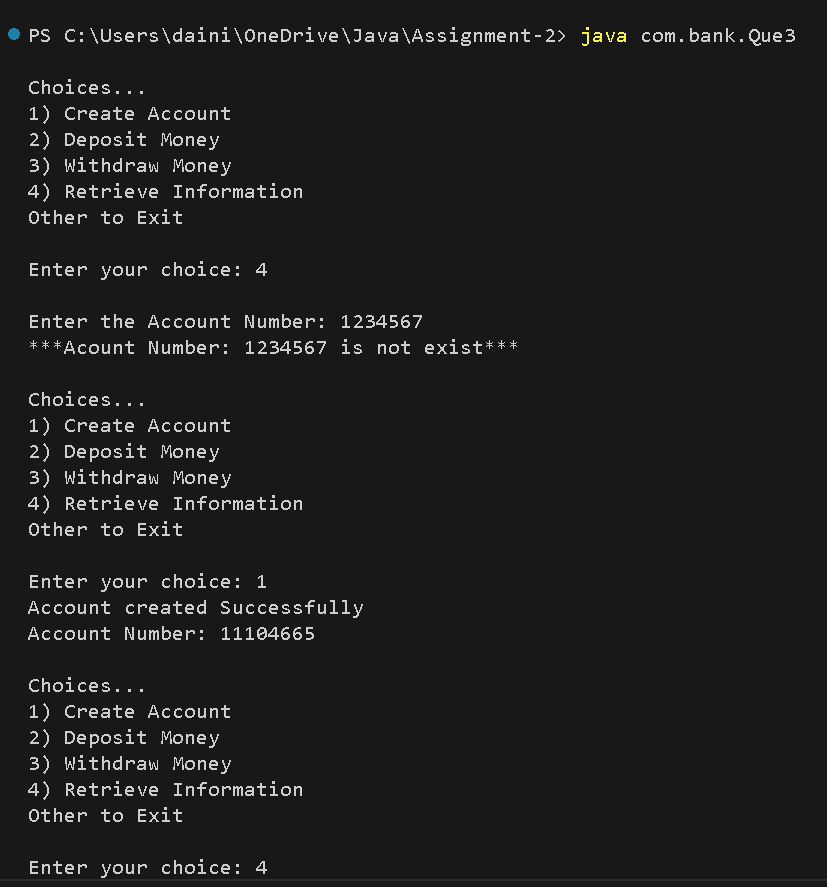
                    }

                    checkAcc = false;

                    break;

                default:

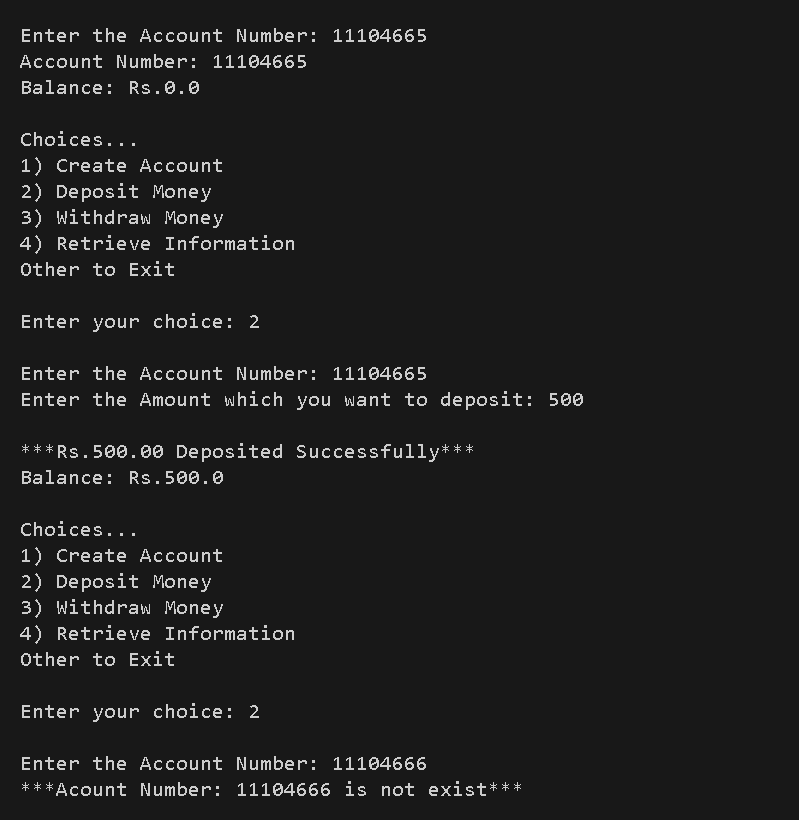
                    return;

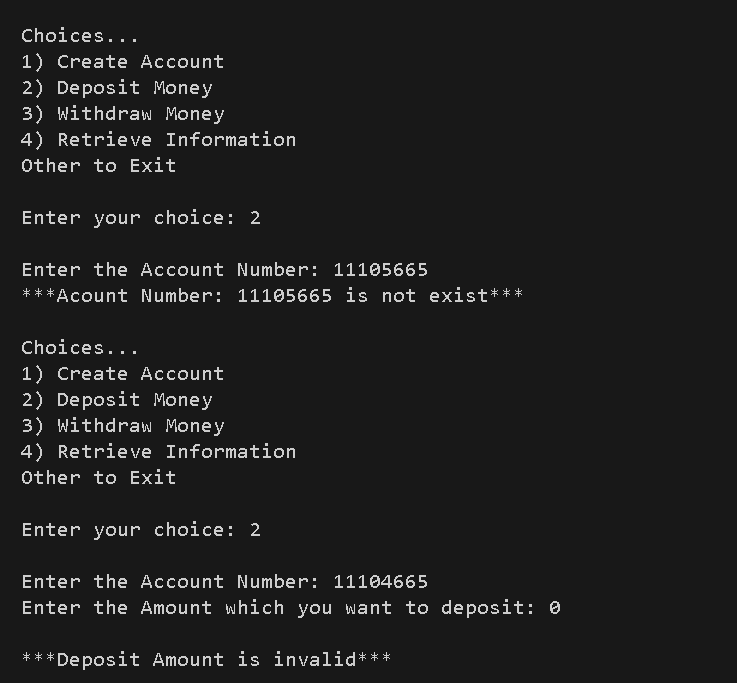
            }

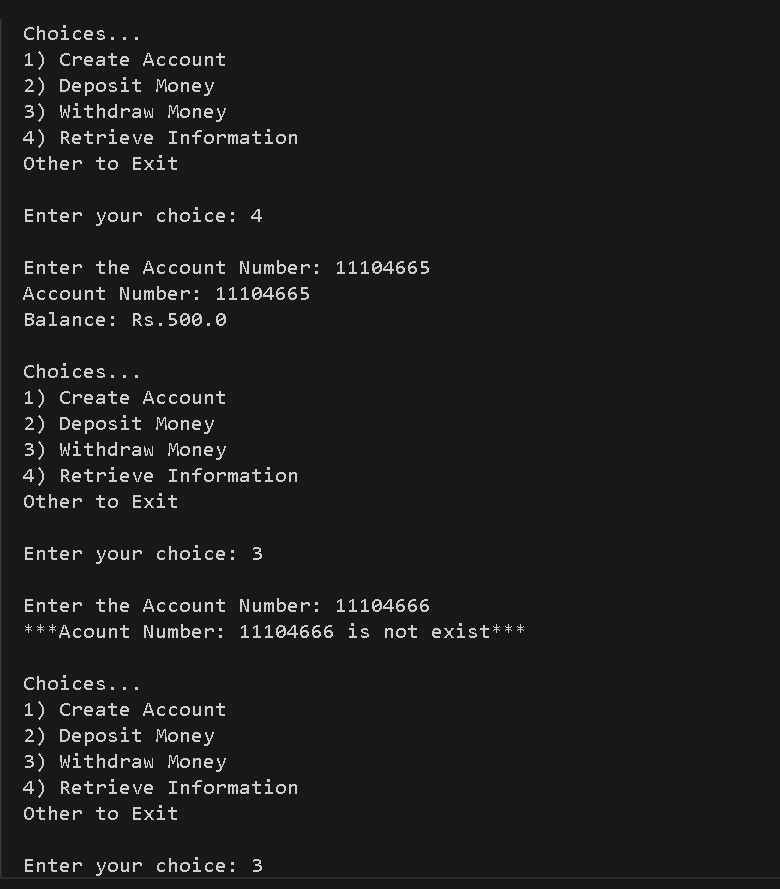
        }

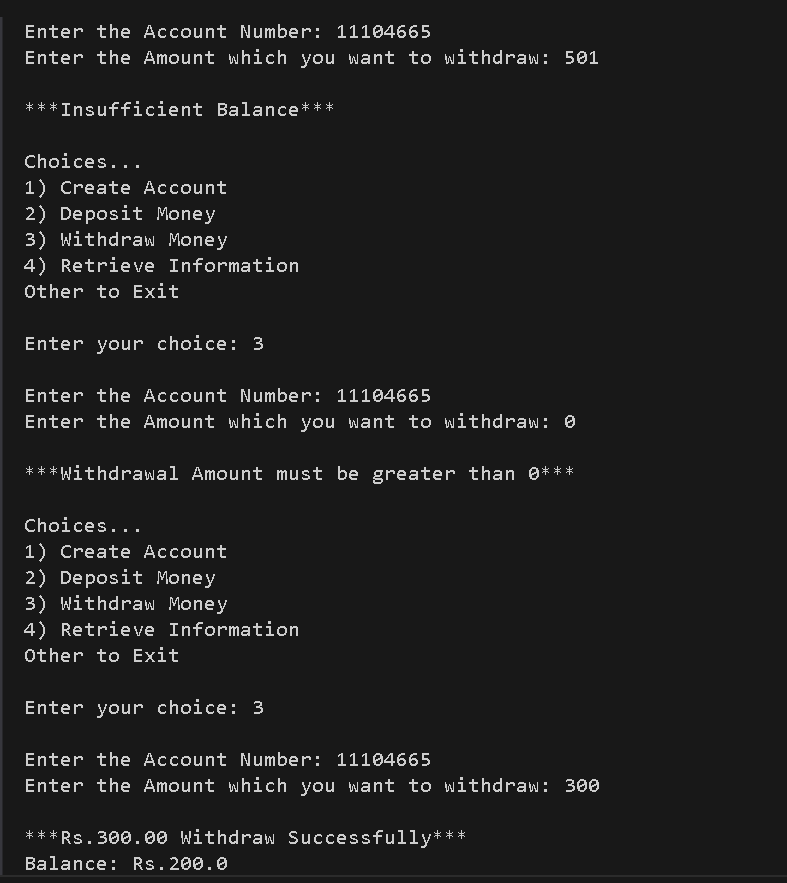
    }

}









**Question-4**

package hospital;

public abstract class Doctor {

    public String name, specialization;

    public Doctor(String name, String specialization) {

        this.name = name;

        this.specialization = specialization;

    }

    public String getName() {

        return this.name;

    }

    public String getSpecialization() {

        return this.specialization;

    }

    public abstract void bookAppointment();

}

class Appointment {

    private Doctor DS;

    Appointment(Doctor DS) {

        this.DS = DS;

    }

    void schedule() {

        System.out.println("\nName: "+DS.getName());

        System.out.println("Specialization: "+DS.getSpecialization());

        DS.bookAppointment();

    }

}

package hospital.doctors;

import hospital.Doctor;

public class Specialist extends hospital.Doctor {

    private String expertise;

    public Specialist(String ...args) {

        super(args[0], args[1]);

        this.expertise = args[2];

    }

    public void bookAppointment() {

        System.out.println("Expertise: "+this.expertise);

        System.out.println("\nYour Appoitment is booked\n");

    }

}

package hospital;

import hospital.doctors.Specialist;

public class Que4 {

    public static void main(String[] args) {

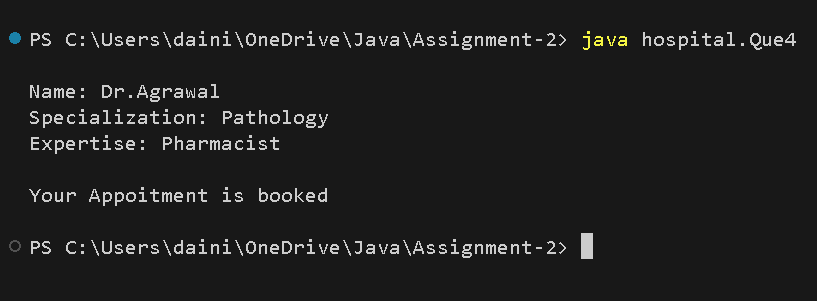
        Specialist S = new Specialist("Dr.Agrawal","Pathology","Pharmacist");

        Appointment A = new Appointment(S);

        A.schedule();

    }

}



**Question-5**

import java.util.List;

import java.util.ArrayList;

import java.util.Scanner;

class BankAccount {

    private String accountHolder;

    private double balance;

    List<String> transactions = new ArrayList<String>();

    BankAccount(String accountHolder) {

        this.accountHolder = accountHolder;

        this.balance = 0;

    }

    void deposit(double amount) {

        if(amount > 0) {

            this.balance += amount;

            System.out.printf("\*\*\*Rs.%.2f is Deposited Successfully\*\*\*\n",amount);

            transactions.add("Deposited Rs."+amount+"\tBalance: Rs."+this.balance);

        }

        else {

            System.out.println("\*\*\*Deposit amount must be greater than 0\*\*\*\n");

        }

    }

    void withdraw(double amount) {

        if(amount <= 0) {

            System.out.println("\*\*\*Withdrawn Amount must be greater than 0\*\*\*\n");

        }

        else if(amount>this.balance) {

            System.out.println("\*\*\*Insuficient Balance\*\*\*\n");

        }

        else {

            this.balance -= amount;

            System.out.printf("\nRs.%.2f is Withdrawn Successfully\n",amount);

            transactions.add("Withdraw Rs."+amount+"\tBalance: Rs."+this.balance);

        }

    }

    void printMiniStatement() {

        System.out.println("\n\nAccount Holder Name: "+this.accountHolder);

        try {

            transactions.get(0);

            System.out.println("\n\tTransaction History");

            System.out.println("-------------------------------------------");

            for(int i=0; i<transactions.size(); i++) {

                System.out.printf(transactions.get(i)+"\n");

            }

        }

        catch(Exception e) {

            System.out.println("\n\*\*\*No Transaction History\*\*\*");

        }

    }

    double getBalance() {

        return this.balance;

    }

}

class Que5 {

    public static void main(String[] args) {

        Scanner S = new Scanner(System.in);

        System.out.print("\nEnter Account Holder Name: ");

        String name = S.nextLine();

        double amount;

        BankAccount B = new BankAccount(name);

        while(true) {

            System.out.println("\nChoices...");

            System.out.println("1) Deposit Money");

            System.out.println("2) Withdraw Money");

            System.out.println("3) Check Balance");

            System.out.println("4) Print Mini Statement");

            System.out.println("-- Other to Exit");

            System.out.print("\nEnter your Choice: ");

            int choice = S.nextInt();

            switch(choice) {

                case 1:

                    System.out.print("\nEnter the Amount to Deposit: ");

                    amount = S.nextDouble();

                    B.deposit(amount);

                    break;

                case 2:

                    System.out.print("\nEnter the Amount to Withdraw: ");

                    amount = S.nextDouble();

                    B.withdraw(amount);

                    break;

                case 3:

                    System.out.println("\nBalance: Rs."+B.getBalance()+"\n");

                    break;

                case 4:

                    B.printMiniStatement();

                    break;

                default:

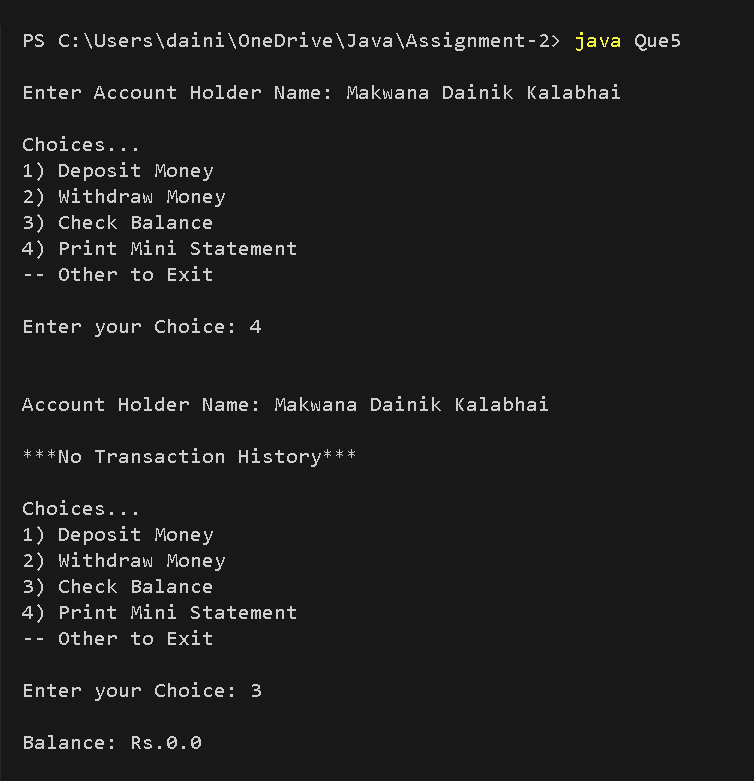
                    return;

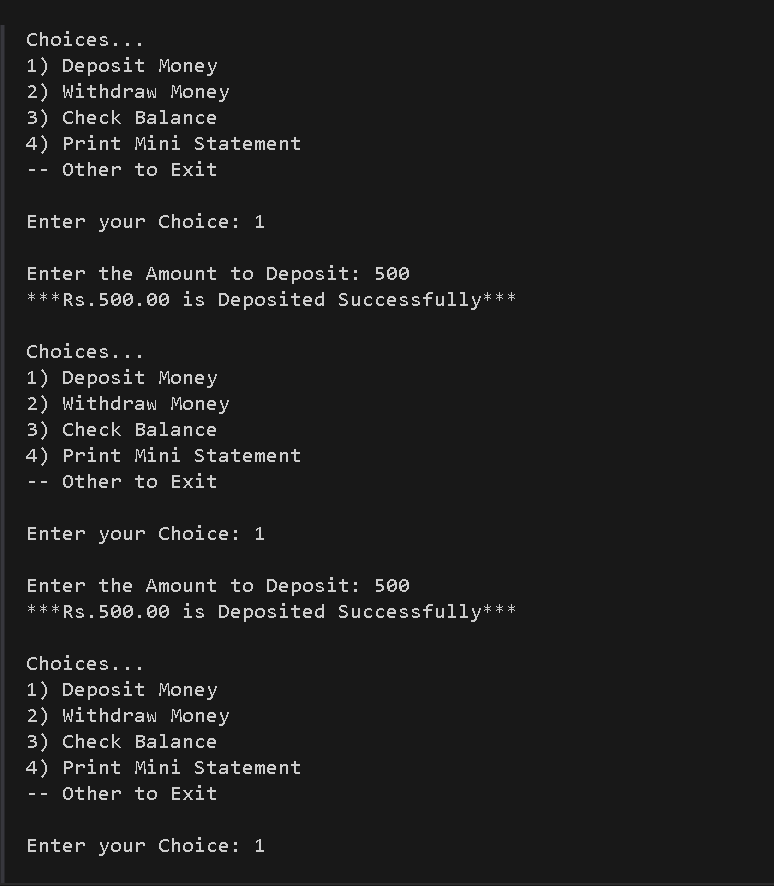
            }

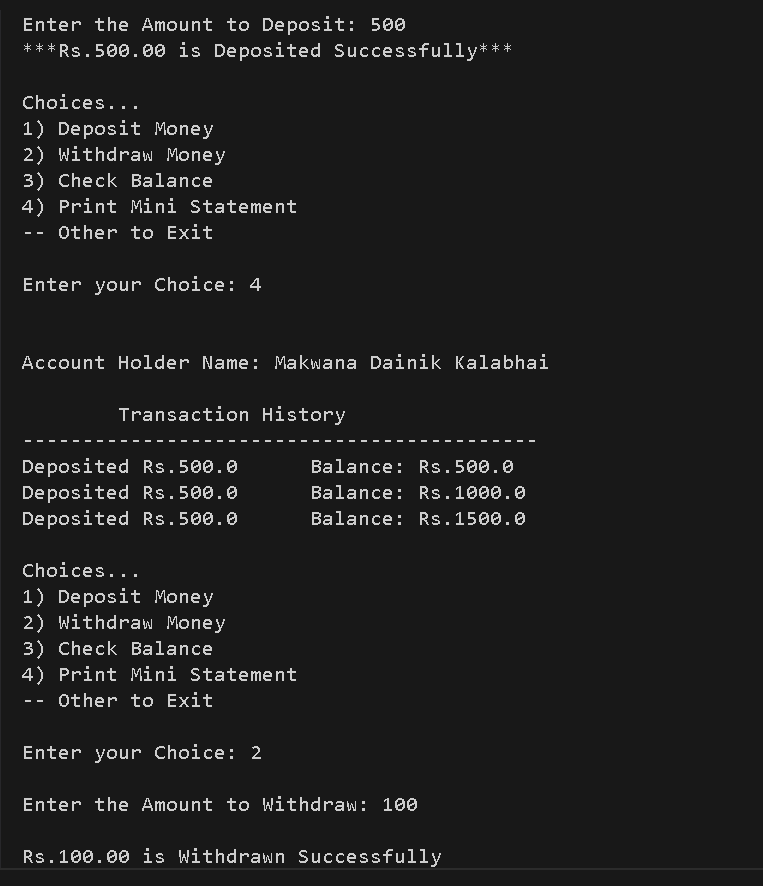
        }

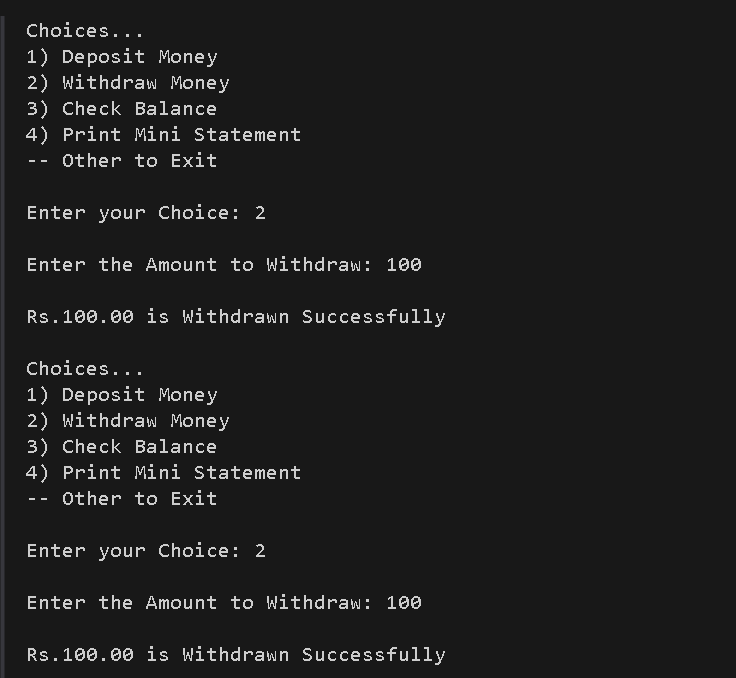
    }

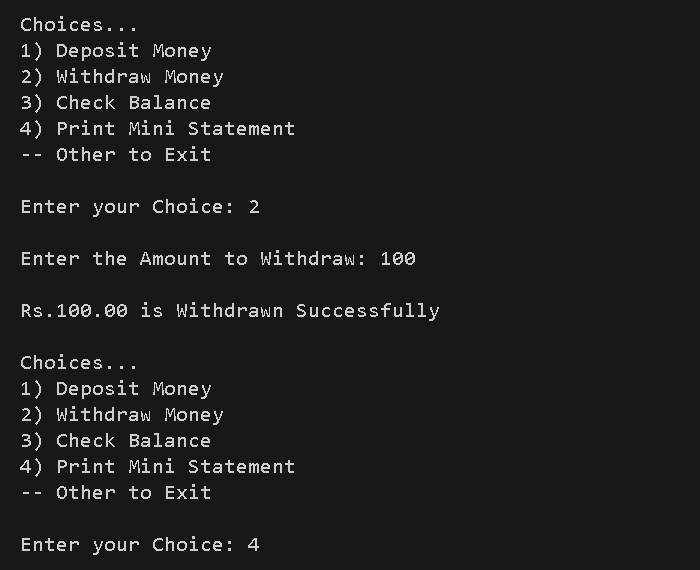
}

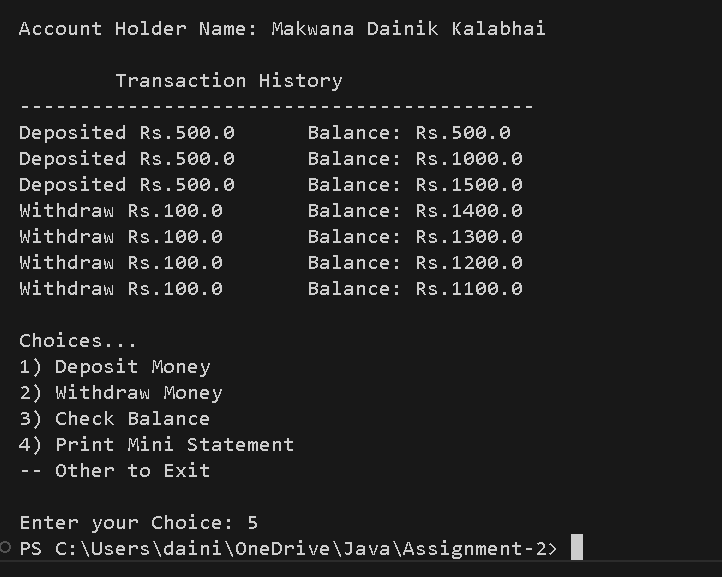












**Question-6**

package propertymanagement;

class Apartments implements Property {

    String location;

    double price;

    Apartments(String location, double price) {

        this.location = location;

        this.price = price;

    }

    public void buy() {

        System.out.println("\n\*\*\*Apartment is buy successfully with\*\*\*\n  Price: Rs."+this.price+"\n  Location: "+this.location);

    }

    public void sell() {

        System.out.println("\n\*\*\*Apartment is sold successfully with\*\*\*\n  Price: Rs."+this.price+"\n  Location: "+this.location);

    }

}

package propertymanagement;

class Bungalow implements Property {

    String location;

    double price;

    Bungalow(String location, double price) {

        this.location = location;

        this.price = price;

    }

    public void buy() {

        System.out.println("\n\*\*\*Bungalow is buy successfully with\*\*\*\n  Price: Rs."+this.price+"\n  Location: "+this.location);

    }

    public void sell() {

        System.out.println("\n\*\*\*Bungalow is sold successfully with\*\*\*\n  Price: Rs."+this.price+"\n  Location: "+this.location);

    }

}

package propertymanagement;

class Tenaments implements Property {

    String location;

    double price;

    Tenaments(String location, double price) {

        this.location = location;

        this.price = price;

    }

    public void buy() {

        System.out.println("\n\*\*\*Tenament is buy successfully with\*\*\*\n  Price: Rs."+this.price+"\n  Location: "+this.location);

    }

    public void sell() {

        System.out.println("\n\*\*\*Tenament is sold successfully with\*\*\*\n  Price: Rs."+this.price+"\n  Location: "+this.location);

    }

}

package propertymanagement;

class Que6 {

    public static void main(String[] args) {

        Property P[] = new Property[6];

        P[0] = new Apartments("Green City, Gandhinagar",500000);

        P[1] = new Bungalow("Haven, New Ranip", 2255000);

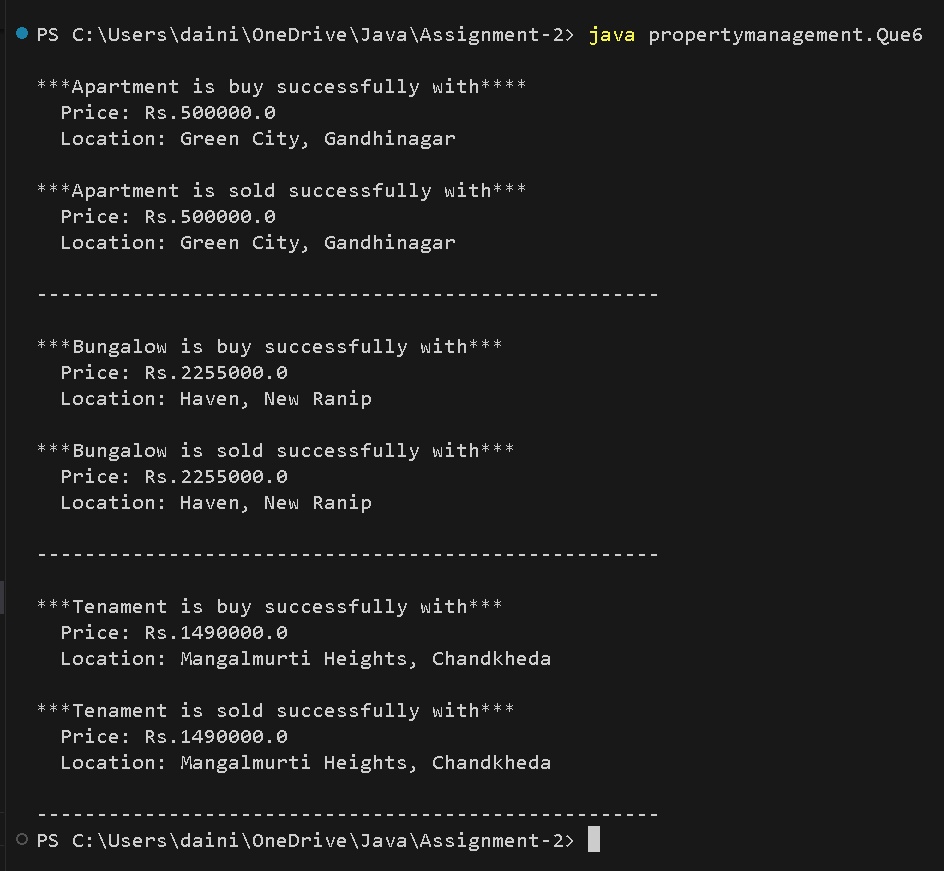
        P[2] = new Tenaments("Mangalmurti Heights, Chandkheda", 1490000);

        for(int i=0; i<3; i++) {

            P[i].buy();

            P[i].sell();

            System.out.println("\n----------------------------------------------------");

        }

    }

}

**Question-7**

class Flour {

    private double weight, price;

    Flour(double ...args) {

        this.weight = args[0];

        this.price = args[1];

    }

    double getWeight() {

        return this.weight;

    }

    double getPrice() {

        return this.price;

    }

    double calcPrice() {

        return weight\*price;

    }

}

class FlourItem {

    Flour defaultFlour() {

        return new Flour(1, 50);

    }

}

interface FlourItemInterface {

    Flour getQuintal();

    Flour get10kg();

    Flour get1kg();

}

class FlourStore implements FlourItemInterface {

    public Flour getQuintal() {

        return new Flour(100, 45);

    }

    public Flour get10kg() {

        return new Flour(10, 48);

    }

    public Flour get1kg() {

        return new Flour(1, 50);

    }

}

class Que7 {

    public static void main(String[] args) {

        FlourItemInterface FS = new FlourStore();

        System.out.println("\n\nFlour Packet (Weight: "+FS.getQuintal().getWeight()+"KG)");

        System.out.println("----------------------------------------");

        System.out.println("Price per KG: Rs."+FS.getQuintal().getPrice());

        System.out.println("\nTotal Price: Rs."+FS.getQuintal().calcPrice());

        System.out.println("\n\nFlour Packet (Weight: "+FS.get10kg().getWeight()+"KG)");

        System.out.println("----------------------------------------");

        System.out.println("Price per KG: Rs."+FS.get10kg().getPrice());

        System.out.println("\nTotal Price: Rs."+FS.get10kg().calcPrice());

        System.out.println("\n\nFlour Packet  (Weight: "+FS.get1kg().getWeight()+"KG)");

        System.out.println("----------------------------------------");

        System.out.println("\nTotal Price: Rs."+FS.get1kg().calcPrice()+"\n\n");

    }

}



**Question-8**

package library;

import java.util.Scanner;

import java.util.ArrayList;

abstract class AbstractBook {

    private String title, author;

    String bookType;

    boolean isLent;

    AbstractBook(String ...a) {

        this.title = a[0];

        this.author = a[1];

        this.bookType = a[2];

        this.isLent = false;

    }

    String getTitle() {

        return this.title;

    }

    String getAuthor() {

        return this.author;

    }

    abstract String getBookType();

    public String toString() {

        return ("Title: "+this.title+"\nAuthor: "+this.author+"\nBook Type: "+this.bookType);

    }

}

class Book extends AbstractBook {

    Book(String ...a) {

        super(a);

    }

    String getBookType() {

        return this.bookType;

    }

}

class Library {

    ArrayList<Book> B = new ArrayList<Book>();

    Library() {

        B.add(new Book("Head First Java","Kathy Sierra","Learning"));

        B.add(new Book("Clean Code","Robert Cecil Martin","Learning"));

        B.add(new Book("A Book On C, 4/E","Kelley","Learning"));

        B.add(new Book("A Tour of C++","Bjarne Stroustrup","Learning"));

        B.add(new Book("A Smarter Way to Learn JavaScript","Mark Myers","Learning"));

        B.add(new Book("HTML & CSS: Design and Build Web Sites","Jon Duckett","Learning"));

        B.add(new Book("React and React Native","Adam Boduch","Learning"));

    }

    void addBook(String ...a) {

        B.add(new Book(a[0], a[1], a[2]));

        System.out.println("\n\*\*\*Book added Succefully\*\*\*");

        System.out.println(B.get(B.size()-1).toString());

    }

    void lendBook(int num) {

        for(int i=0; i<B.size(); i++) {

            if((i+1) == num) {

                B.get(i).isLent = true;

                System.out.println("\n\*\*\*Book Lended Successfully\*\*\*");

                System.out.println(B.get(i).toString());

                return;

            }

        }

    }

    void returnBook(int num) {

        for(int i=0; i<B.size(); i++) {

            if((i+1) == num) {

                B.get(i).isLent = false;

                System.out.println("\n\*\*\*Book Return Successfully\*\*\*");

                System.out.println(B.get(i).toString());

                return;

            }

        }

    }

    void printAvBook() {

        int count = 0;

        for(int i=0; i<B.size(); i++) {

            if(!B.get(i).isLent) {

                count++;

                break;

            }

        }

        if(count > 0) {

            System.out.println("\nAvailable Books");

            System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

            for(int i=0; i<B.size(); i++) {

                if(!B.get(i).isLent) {

                    System.out.println((i+1)+") "+B.get(i).toString());

                    System.out.println("-----------------------------------------------");

                    count++;

                }

            }

        }

        else {

            System.out.println("\*\*\*All Books are Lended Now\*\*\*");

        }

    }

    void printNotAvBook() {

        int count = 0;

        for(int i=0; i<B.size(); i++) {

            if(B.get(i).isLent) {

                count++;

                break;

            }

        }

        if(count > 0) {

            System.out.println("\nAll Lended Books");

            System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

            for(int i=0; i<B.size(); i++) {

                if(B.get(i).isLent) {

                    System.out.println((i+1)+") "+B.get(i).toString());

                    System.out.println("-----------------------------------------------");

                    count++;

                }

            }

        }

        else {

            System.out.println("\*\*\*All Books are Available, No Any Book is Lended Now\*\*\*");

        }

    }

}

class Que8 {

    public static void main(String[] args) {

        Scanner S = new Scanner(System.in);

        Library L = new Library();

        int choice, num;

        String title, author, bookType;

        while(true) {

            System.out.println("\n\nChoices....");

            System.out.println("1) Add Book in Library");

            System.out.println("2) Lend Book");

            System.out.println("3) Return Book");

            System.out.println("4) List All Available Books");

            System.out.println("5) List All lended Books");

            System.out.println("-- Other to Exit");

            System.out.print("\nEnter Your Choice: ");

            choice = S.nextInt();

            switch(choice) {

                case 1:

                    Scanner S1 = new Scanner(System.in);

                    System.out.print("Enter Title of the Book: ");

                    title = S1.nextLine();

                    Scanner S2 = new Scanner(System.in);

                    System.out.print("Enter Author of the Book: ");

                    author = S2.nextLine();

                    Scanner S3 = new Scanner(System.in);

                    System.out.print("Enter Book Type: ");

                    bookType = S3.nextLine();

                    L.addBook(title, author, bookType);

                    break;

                case 2:

                    L.printAvBook();

                    System.out.print("\nEnter the Number of the Book, Which you want to Lend: ");

                    num = S.nextInt();

                    L.lendBook(num);

                    break;

                case 3:

                    L.printNotAvBook();

                    System.out.print("\nEnter the Number of the Book, Which you want to Return: ");

                    num = S.nextInt();

                    L.returnBook(num);

                    break;

                case 4:

                    L.printAvBook();

                    break;

                case 5:

                    L.printNotAvBook();

                    break;

                default:

                    return;

            }

        }

    }

}

