Subhradip Debnath
Sec : A
Roll : 19
CSE Department
Institute of Engineering and Management, Kolkata
Date : Wednesday, 4 November 2020

Assignment 8

Question 1 : Pay

Code :

---

pay.java

```java
import java.util.Scanner;

public class pay {
    private double hoursWorked;
    private double payRate;
    private double withHoldingRate;
    private double grossPay;
    private double netPay;

    double computeNetPay(double hoursWorked){
        this.hoursWorked = hoursWorked;
        this.payRate = 5.85;
        this.grossPay = 5.85 * hoursWorked;
        this.withHoldingRate = 0.15;
        this.netPay = this.grossPay - (this.grossPay * 0.15);
        return this.netPay;
    }
    double computeNetPay(double hoursWorked, double payRate){
        this.hoursWorked = hoursWorked;
        this.payRate = payRate;
        this.grossPay = payRate * hoursWorked;
        this.withHoldingRate = 0.15;
        this.netPay = this.grossPay - (this.grossPay * 0.15);
        return this.netPay;
    }
    double computeNetPay(double hoursWorked, double payRate, double withHoldingRate){
        this.hoursWorked = hoursWorked;
        this.payRate = payRate;
        this.grossPay = payRate * hoursWorked;
        this.withHoldingRate = withHoldingRate;
        this.netPay = this.grossPay - (this.grossPay * withHoldingRate);
        return this.netPay;
    }

    public static void main(String[] args) {
        double hrs, rate, withHolding;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter worked hours:");
        hrs = sc.nextDouble();
        System.out.println("Enter rate:");
```

```java
        rate = sc.nextDouble();
        System.out.println("Enter with holding rate:");
        withHolding = sc.nextDouble();

        pay p1 = new pay();

        System.out.println("Using only one parameter: " + p1.computeNetPay(hrs));
        System.out.println("Using only two parameter: " + p1.computeNetPay(hrs,rate));
        System.out.println("Using only three parameter: " +
p1.computeNetPay(hrs,rate,withHolding));
    }

}
```

Output :

```
Enter worked hours:
100
Enter rate:
5.85
Enter with holding rate:
0.15
Using only one parameter: 497.25
Using only two parameter: 497.25
Using only three parameter: 497.25
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Question 2 : Hotel
Code :

HotelRoom.java

```java
public class HotelRoom {
    int roomNo;
    double rentalRate;
    HotelRoom(int roomNo){
        this.roomNo = roomNo;
        if(this.roomNo <= 299 ) {
        this.rentalRate = 69.95;
        }
        else {
                this.rentalRate = 89.95;
        }
    }

    int getRoomNo(){
        return this.roomNo;
    }
    double getRentalRate() {
        return this.rentalRate;
    }
}
```

Suite.java

```java
public class Suite extends HotelRoom{
    Suite(int roomNo){
        super(roomNo);
        if(super.roomNo <= 299) {
                super.rentalRate = super.rentalRate + 40;
        }
        else {
                super.rentalRate = super.rentalRate + (super.rentalRate * 0.15) + 40;
        }
    }
}
```

UseHotelRoom.java

```java
import java.util.*;
public class UseHotelRoom {
    public static void main(String [] args) {
                int smallRoom;
                int bigRoom;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the small Room no: ");
                smallRoom = sc.nextInt();
                System.out.println("Enter the big Room no: ");
                bigRoom = sc.nextInt();

                HotelRoom h1 = new HotelRoom(smallRoom);
```

```
            System.out.println("Hotel Room No: " + h1.getRoomNo() + " Rental Rate: " +
h1.getRentalRate());

            HotelRoom h2 = new HotelRoom(bigRoom);
            System.out.println("Hotel Room No: " + h2.getRoomNo() + " Rental Rate: " +
h2.getRentalRate());

            Suite s1 = new Suite(smallRoom);
            System.out.println("Hotel Suite Room No: " + s1.getRoomNo() + " Rental Rate: " +
s1.getRentalRate());

            Suite s2 = new Suite(bigRoom);
            System.out.println("Hotel Suite Room No: " + s2.getRoomNo() + " Rental Rate: " +
s2.getRentalRate());
    }
}
```

Output :

```
Enter the small Room no:
100
Enter the big Room no:
500
Hotel Room No: 100 Rental Rate: 69.95
Hotel Room No: 500 Rental Rate: 89.95
Hotel Suite Room No: 100 Rental Rate: 109.95
Hotel Suite Room No: 500 Rental Rate: 143.4425
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Question 3 : Order
Code :

Order.java

```java
public class Order {
    private String customerName;
    private int customerNo;
    private int quantityOrdered;
    private int unitPrice;
    private int totalPrice;

    public Order(String customerName, int customerNo, int quantityOrdered, int unitPrice) {
        this.customerName = customerName;
        this.customerNo = customerNo;
        this.quantityOrdered = quantityOrdered;
        this.unitPrice = unitPrice;
    }

    public String getCustomerName() {
        return customerName;
    }

    public int getCustomerNo() {
        return customerNo;
    }

    public int getQuantityOrdered() {
        return quantityOrdered;
    }

    public int getUnitPrice() {
        return unitPrice;
    }


    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public void setCustomerNo(int customerNo) {
        this.customerNo = customerNo;
    }

    public void setQuantityOrdered(int quantityOrdered) {
        this.quantityOrdered = quantityOrdered;
    }

    public void setUnitPrice(int unitPrice) {
        this.unitPrice = unitPrice;
    }

    public int computePrice(int quantity){
        this.quantityOrdered = quantity;
        this.totalPrice = this.quantityOrdered * this.unitPrice;
        return this.totalPrice;
    }
```

}

ShippedOrder.java

```java
public class ShippedOrder extends Order{

    public ShippedOrder(String customerName, int customerNo, int quantityOrdered, int unitPrice) {
        super(customerName, customerNo, quantityOrdered, unitPrice);
    }

    public int computePrice(int quantity){
        int price = super.computePrice(quantity);
        return price + 4;
    }
}
```

UseOrder.java

```java
public class UseOrder {
    public static void main(String[] args) {
        Order order = new Order("Sam", 34809, 60, 40);
        System.out.println("Customer Number: " + order.getCustomerNo());
        System.out.println("Name: " + order.getCustomerName());
        System.out.println("Quantity Ordered: " + order.getQuantityOrdered());
        System.out.println("Unit Price: " + order.getUnitPrice());
        System.out.println("Total: " + order.computePrice(order.getQuantityOrdered()));

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

        ShippedOrder shippedOrder = new ShippedOrder("John", 34325, 50, 30);
        System.out.println("Customer Number: " + shippedOrder.getCustomerNo());
        System.out.println("Name: " + shippedOrder.getCustomerName());
        System.out.println("Quantity Ordered: " + shippedOrder.getQuantityOrdered());
        System.out.println("Unit Price: " + shippedOrder.getUnitPrice());
        System.out.println("Total: " +
shippedOrder.computePrice(shippedOrder.getQuantityOrdered()));
    }
}
```

Output :

```
Customer Number: 34809
Name: Sam
Quantity Ordered: 60
Unit Price: 40
Total: 2400


Customer Number: 34325
Name: John
Quantity Ordered: 50
Unit Price: 30
Total: 1504
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Question 4 : Account

Code :

CheckingAccount.java

```java
public class CheckingAccount {
    int accountNumber;
    int balance;
    public CheckingAccount(int accountNumber, int balance) {
        this.accountNumber = accountNumber;
        if(balance < 200){
            this.balance = 0;
        }else{
            this.balance = balance;
        }
    }

    public void getAccountDetails(){
        System.out.println("Account Number: " + this.accountNumber);
        if(this.balance == 0){
            System.out.println("Balance: " + this.balance);
            System.out.println("Minimum 200$ is required for account");
        }else{
            System.out.println("Balance: " + this.balance);
        }
    }
}
```

TestAccount.java

```java
import java.util.Scanner;

public class TestAccount {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter account number: ");
        int accountNumber = sc.nextInt();
        System.out.println("Enter balance: ");
        int balance = sc.nextInt();
        CheckingAccount acc = new CheckingAccount(accountNumber, balance);
        acc.getAccountDetails();
    }   }
```

Output :

```
Enter account number:
10
Enter balance:
400
Account Number: 10
Balance: 400
--------------------------------------------------------------------
BUILD SUCCESS
--------------------------------------------------------------------
```