**TAMOJIT DAS A/7/C1/3rd**

**ASSIGNMENT 8**

Q1.

class FoodItem {

private String desc;

private double unitPrice;

public FoodItem(String desc, double unitPrice){

this.desc = desc;

this.unitPrice = unitPrice;

}

public String getDesc(){

return desc;

}

public double getUnitPrice(){

return unitPrice;

}

}

import java.time.\*;

class FoodLineItem {

private FoodItem item;

private int quantity;

private double price;

private LocalDateTime time;

public FoodLineItem(FoodItem item, int quantity, LocalDateTime time){

this.item = item;

this.quantity = quantity;

this.price = quantity \* item.getUnitPrice();

this.time = time;

}

public FoodItem getItem(){

return item;

}

public int getQuantity(){

return quantity;

}

public double getPrice(){

return price;

}

public LocalDateTime getTime(){

return time;

}

}

import java.io.\*;

import java.time.LocalDateTime;

import java.util.\*;

class BillingManager {

private List<FoodLineItem> myList;

private List<String> discCategory;

private List<Double> myDiscount;

public BillingManager(){

myList = new ArrayList<>();

discCategory = new ArrayList<>();

myDiscount = new ArrayList<>();

}

public void addItem(FoodLineItem myItem){

myList.add(myItem);

}

public void discount(){

LocalDateTime t1 = LocalDateTime.of(1,1,1,11,0);

LocalDateTime t2 = LocalDateTime.of(1,1,1,15,0);

double sum = 0;

for(FoodLineItem i : myList){

if(i.getTime().compareTo(t1) >= 0 && i.getTime().compareTo(t2) <= 0)

sum += i.getPrice();

}

if(sum > 0){

discCategory.add("Discount (Lunch-Time Special)");

myDiscount.add(sum \* 0.10);

}

}

public void discount(FoodItem pizza1, FoodItem pizza2){

double a = pizza1.getUnitPrice(), b = pizza2.getUnitPrice();

discCategory.add("Discount (Combo Special)");

myDiscount.add(Math.min(a, b));

}

public void discount(FoodItem hotDrink, FoodItem cake, boolean dummyFlag){

discCategory.add("Discount (Tea-Time Special)");

double a = hotDrink.getUnitPrice() + cake.getUnitPrice();

myDiscount.add(a > 200 ? a - 200 : 0);

}

public void discount(boolean isTakeAway){

double sum = 0.0d;

for(FoodLineItem i : myList){

String t = i.getItem().getDesc();

if(t.substring(t.length() - 11).equalsIgnoreCase("(Take-Away)"))

sum += i.getPrice();

}

discCategory.add("Discount (Take-Away)");

myDiscount.add(sum \* 0.15);

}

public void printReceipt(){

int l = Integer.MIN\_VALUE;

for(FoodLineItem i : myList){

l = Math.max(l, i.getItem().getDesc().length());

}

l += 5;

int c = 0;

double total = 0.0d;

String temp = "Food Item Description";

int sp = (l - temp.length()) / 2;

System.out.printf(" %" + (sp + temp.length()) + "s%" + sp + "s%10s" + "%15s" + "%10s%n", temp, "", "Unit Price", "Quantity", "Price");

for(FoodLineItem i : myList){

c++;

total += i.getPrice();

System.out.printf(c + ". " + "%-" + l + "s%10.2f" + "%15d" + "%10.2f%n", i.getItem().getDesc(), i.getItem().getUnitPrice(), i.getQuantity(), i.getPrice());

}

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

total -= myDiscount.get(i);

System.out.printf(" %" + l + "s%25s%10.2f%n", discCategory.get(i), "", -1 \* myDiscount.get(i));

}

System.out.printf("%" + (l + 13) + "s%-15s%10.2f%n", "", "Total", total);

}

public static void main(String[] args)throws IOException{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

BillingManager bill = new BillingManager();

String temp;

FoodLineItem ob1 = null, ob2 = null, tea1 = null, tea2 = null;

LocalDateTime time1 = LocalDateTime.of(1,1,1,11,0);

LocalDateTime time2 = LocalDateTime.of(1,1,1,15,0);

LocalDateTime time3 = LocalDateTime.of(1,1,1,17,0);

do{

System.out.print("FOOD ITEM DESCRIPTION : ");

temp = br.readLine();

if(temp.equals("STOP"))

break;

System.out.print("UNIT PRICE : ");

double t1 = Double.parseDouble(br.readLine());

System.out.print("QUANTITY : ");

int t2 = Integer.parseInt(br.readLine());

System.out.print("ORDER TIME :-\nHOUR : ");

int hr = Integer.parseInt(br.readLine());

System.out.print("MINUTE : ");

int mint = Integer.parseInt(br.readLine());

LocalDateTime time = LocalDateTime.of(1, 1, 1, hr, mint);

if(temp.substring(0, 5).equalsIgnoreCase("pizza"))

temp += " (Combo Special)";

else if(time1.compareTo(time) <= 0 && time2.compareTo(time) >= 0)

temp += " (Lunch-Time Special)";

else if(time2.compareTo(time) <= 0 && time3.compareTo(time) >= 0)

temp += " (Tea-Time Special)";

else

temp += " (Take-Away)";

FoodLineItem ob = new FoodLineItem(new FoodItem(temp, t1), t2, time);

if((ob1 == null || ob2 == null) && temp.substring(0, 5).equalsIgnoreCase("pizza")){

if(ob1 == null)

ob1 = ob;

else

ob2 = ob;

}

if(ob1 != null && ob2 != null){

bill.discount(ob1.getItem(), ob2.getItem());

ob1 = ob2 = null;

}

if((tea1 == null || tea2 == null) && (temp.indexOf("Cake") != -1 || temp.indexOf("Coffee") != -1)){

if(tea1 == null)

tea1 = ob;

else

tea2 = ob;

}

if(tea1 != null && tea2 != null){

bill.discount(tea1.getItem(), tea2.getItem(), true);

tea1 = tea2 = null;

}

bill.addItem(ob);

}while(true);

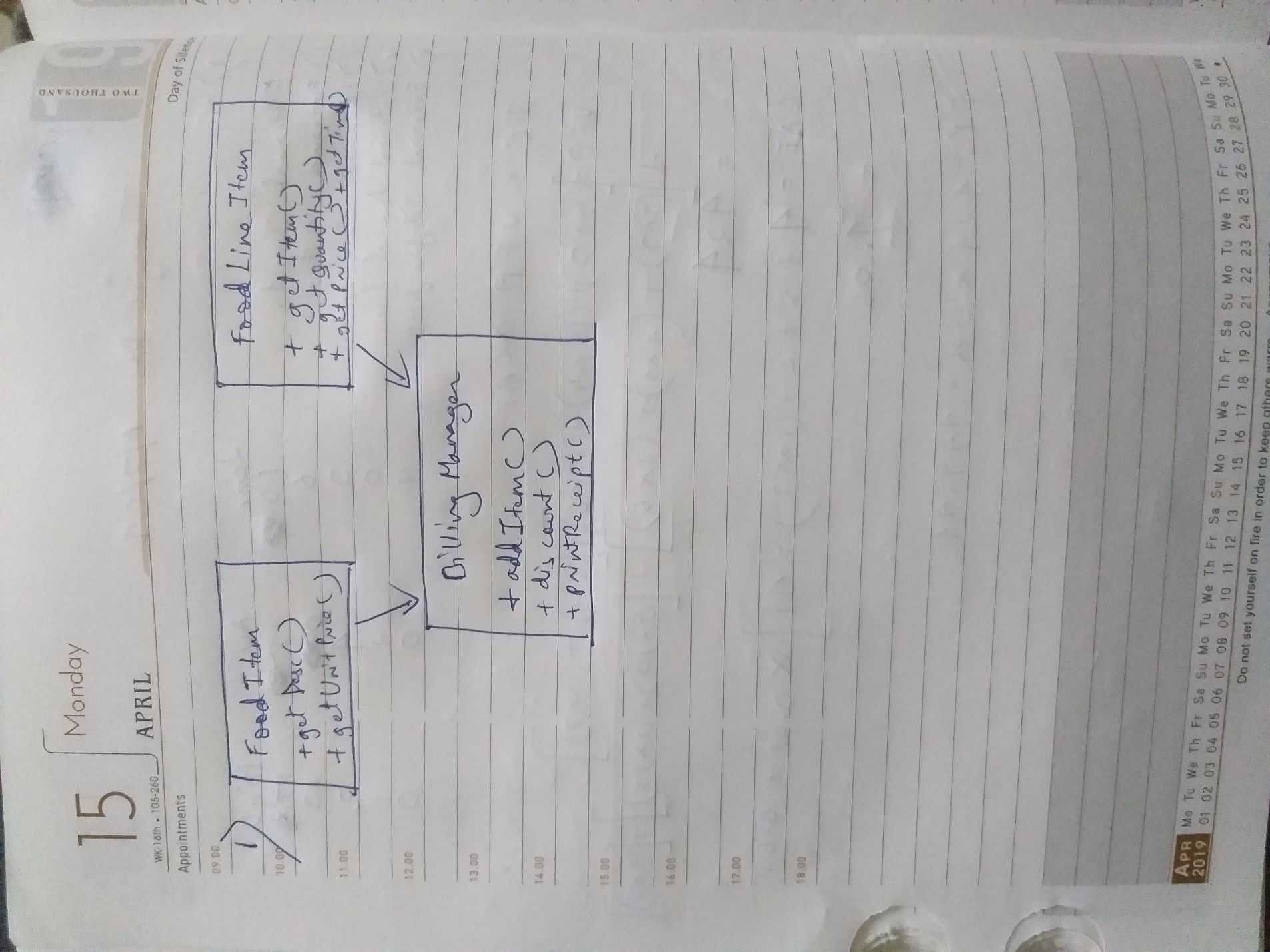
bill.discount();

bill.discount(true);

bill.printReceipt();

}

}



Q2.

import java.util.ArrayList;

class Professor{

String Name;

String Designation;

int Experience;

ArrayList<String> Work;

Professor(String name,int exp){

this.Name=name;

this.Experience=exp;

this.Designation=Apply\_desig(exp);

this.Work=Apply\_work(this.Designation);

}

String Apply\_desig(int exp){

if(exp<5){

return "Assistant Professor";

}else if(exp>=5 && exp<10){

return "Associate Professor";

}else{

return "HOD";

}

}

ArrayList<String> Apply\_work(String desig){

ArrayList<String> work = new ArrayList<String>();

work.add("I am teaching.");

work.add("I do research.");

if(desig=="Associate Professor"){

work.add("I am heading a research group.");

}else if(desig=="HOD"){

work.add("I am heading a research group.");

work.add("I am managing the dept.");

}

return work;

}

void Display(){

String s="Name: "+this.Name+"\nDesignation: "+this.Designation+"\nExp: "+this.Experience+" years\nWork:\n";

for(String each:this.Work){

s=s+each+"\n";

}

System.out.println(s);

}

}

class HRManager{

public static void main(String[] args) {

Professor P1=new Professor("A",4);

Professor P2=new Professor("B",7);

Professor P3=new Professor("C",12);

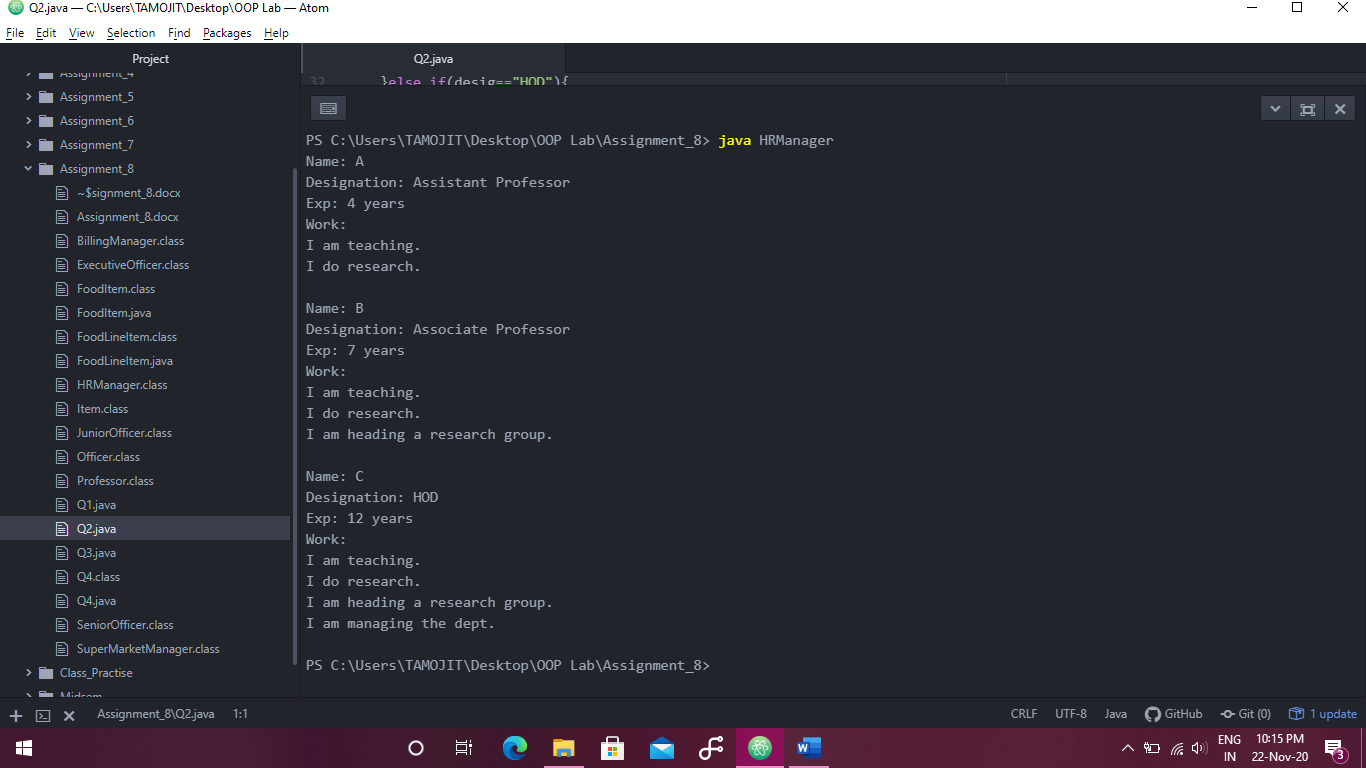
P1.Display();

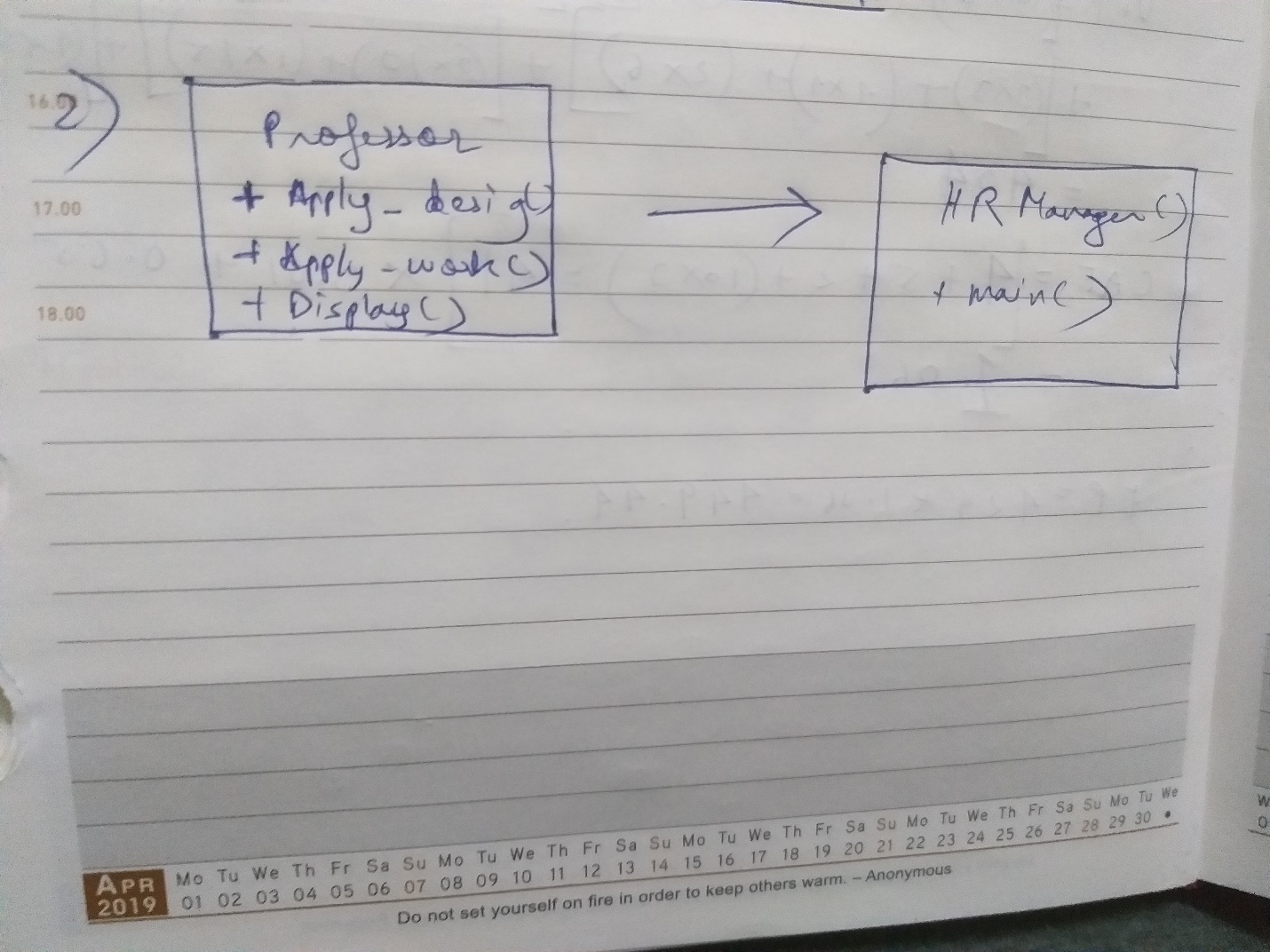
P2.Display();

P3.Display();

}

}





Q3.

import java.util.\*;

class Item{

String Name;

float Qty,Price,Total;

Item(String name,float q,float p){

this.Name=name;

this.Qty=q;

this.Price=p;

this.Total=q\*p;

}

void Display(){

String s="Name: "+this.Name+"\tQuantity: "+this.Qty+"\tPrice: "+this.Price+"\tTotal: "+this.Total;

System.out.println(s);

}

}

class SuperMarketManager{

static ArrayList<Item> ItemList=new ArrayList<Item>();

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

static{

ItemList.clear();

}

public static void main(String[] args) {

AddItem();

AddItem();

AddItem();

Display();

DelItem();

Display();

DelItem();

Display();

CheckOut();

}

static void AddItem(){

System.out.println("Item Name:");

String name=Sc.next();

System.out.println("Quantity");

float qty=Sc.nextFloat();

System.out.println("Price");

float p=Sc.nextFloat();

Item item=new Item(name,qty,p);

ItemList.add(item);

System.out.println("Item Added");

}

static void Display(){

for(Item item:ItemList){

item.Display();

}

}

static void DelItem(){

System.out.println("Enter Index");

int index=Sc.nextInt();

try{

ItemList.remove(index);

}catch(Exception ex){

System.out.println("Item Not Found");

}

}

static void CheckOut(){

float GrandTotal=0.0f;

for(Item item:ItemList){

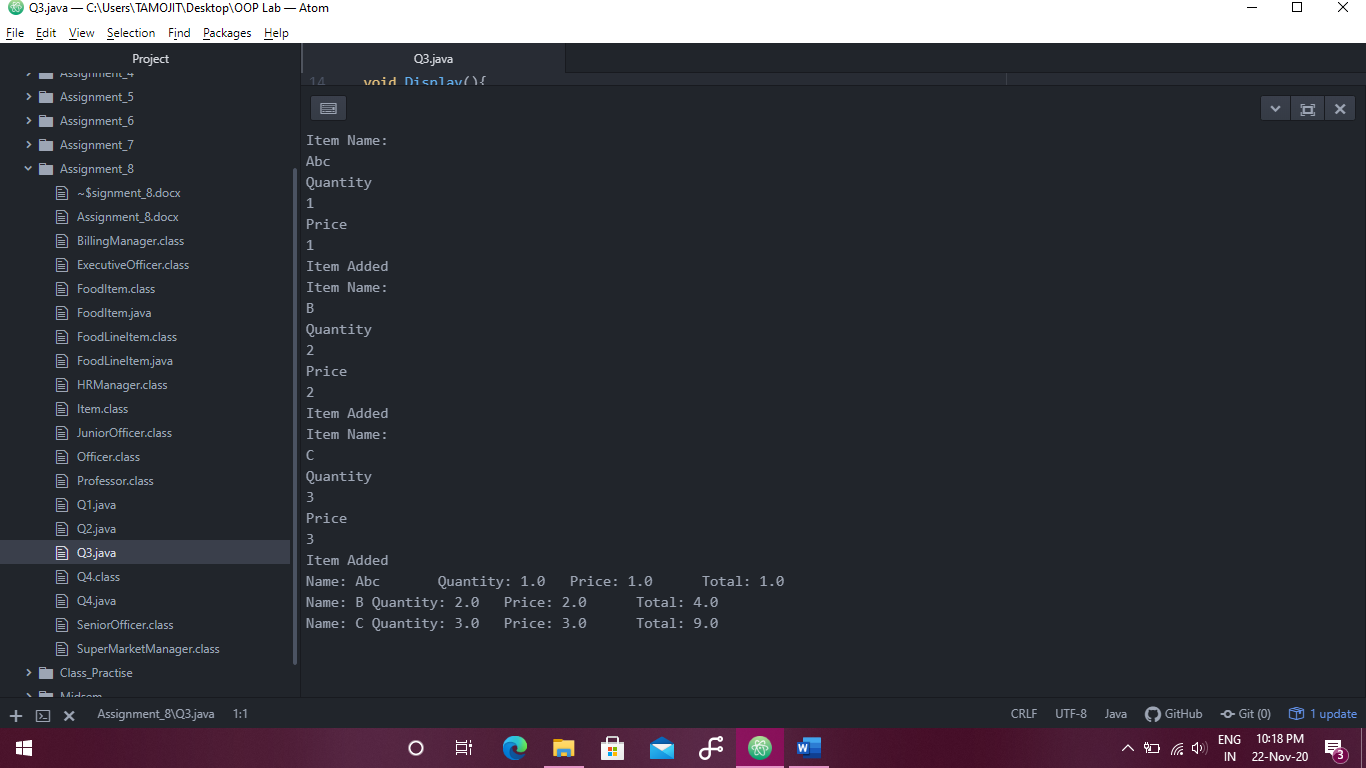
GrandTotal+=item.Total;

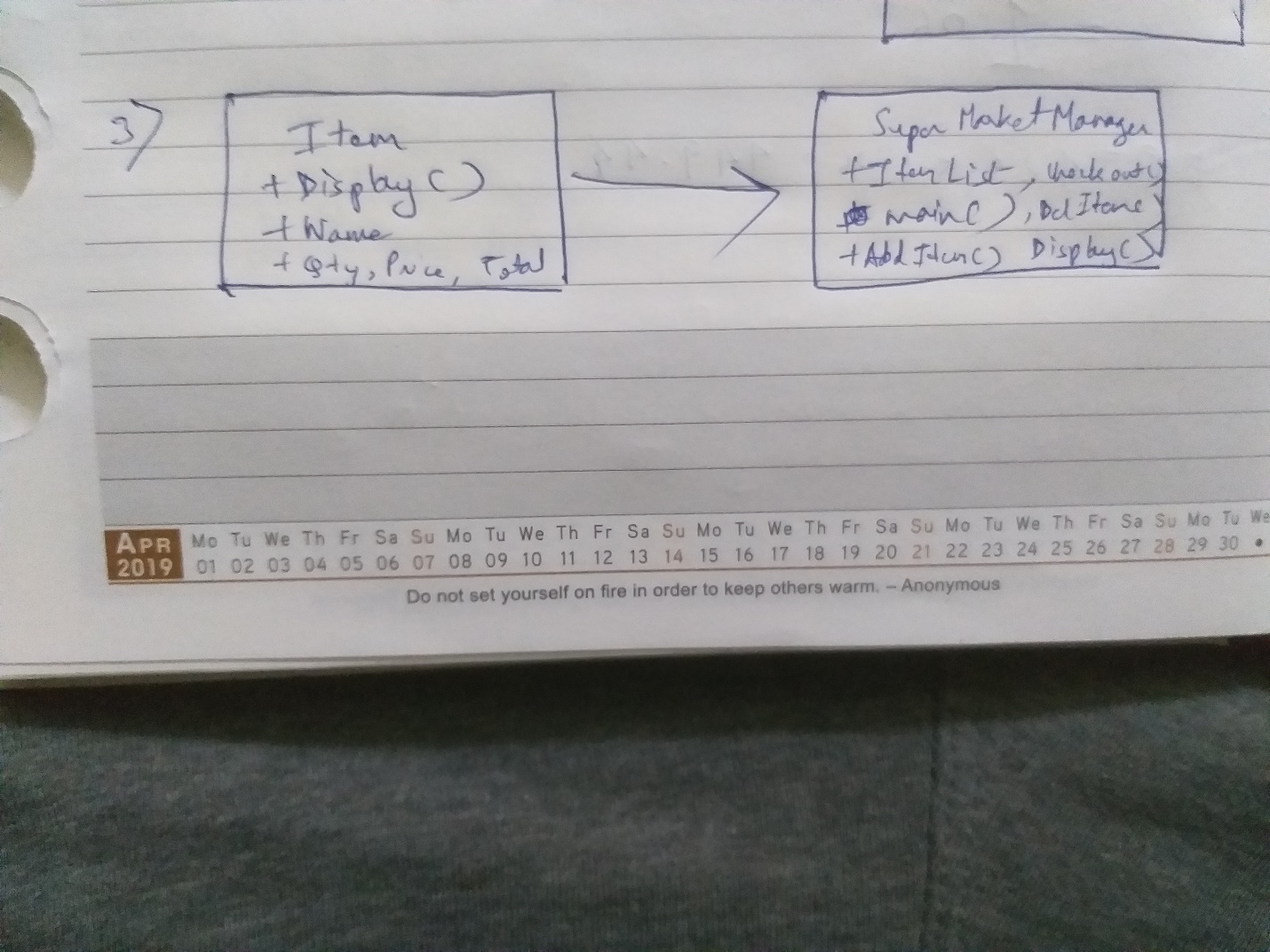
}

System.out.println("GrandTotal: "+GrandTotal);

}

}





Q4.

class Officer{

String Name;

double Limit;

Officer(String name,double limit){

this.Name=name;

this.Limit=limit;

}

void MoneyApproveText(float money){

System.out.println("Approved: "+money);

}

void DontApproveText(){

System.out.println("Not Approved: ");

}

void Approve(float money){

if(money<=this.Limit){

MoneyApproveText(money);

}else{

DontApproveText();

}

}

void Display(){

System.out.println("Name: "+this.Name+"\tLimit: "+this.Limit);

}

}

class JuniorOfficer extends Officer{

JuniorOfficer(String name){

super(name,10000);

}

}

class SeniorOfficer extends Officer{

SeniorOfficer(String name){

super(name,50000);

}

}

class ExecutiveOfficer extends Officer{

ExecutiveOfficer(String name){

super(name,100000);

}

}

class Q4{

public static void main(String[] args) {

JuniorOfficer J=new JuniorOfficer("J");

SeniorOfficer S=new SeniorOfficer("S");

ExecutiveOfficer E=new ExecutiveOfficer("E");

J.Display();

S.Display();

E.Display();

J.Approve(5000);

J.Approve(55000);

S.Approve(5000);

J.Approve(55000);

E.Approve(55000);

E.Approve(5000000);

}

}

