**Question 1: (2 points)**

Design and code a class named **ProjectMember** that holds information of a ProjectMember:

ProjectMember(projectid: int, employee:String, isFullTime:boolean, hours:int)

The program Menu:

1. TC = 1 - test getProjectid()

2. TC = 2 - test setHours()

3. TC = 3 - test setIsFullTime()

4. TC = 4 - test getEmployee()

5. TC = 5 - test toString()

|  |  |
| --- | --- |
| TEST CASE 1  INPUT:  Enter Project ID: 1234  Enter Employee: To An An  Enter IsfullTime (true or false): true  Enter Hours: 120  1. TC = 1 - test getProjectid()  2. TC = 2 - test setHours()  3. TC = 3 - test setIsFullTime()  4. TC = 4 - test getEmployee()  5. TC = 5 - test toString()Enter  TC: 1  OUTPUT:  1234 | TEST CASE 2  INPUT:  Enter Project ID: 1234  Enter Employee: To An An  Enter IsfullTime (true or false): true  Enter Hours: 120  1. TC = 1 - test getProjectid()  2. TC = 2 - test setHours()  3. TC = 3 - test setIsFullTime()  4. TC = 4 - test getEmployee()  5. TC = 5 - test toString()Enter  TC: 2  Enter new Hours: 230  OUTPUT:  230 |
| TEST CASE 3  INPUT:  Enter Project ID: 1234  Enter Employee: To An An  Enter IsfullTime (true or false): true  Enter Hours: 120  1. TC = 1 - test getProjectid()  2. TC = 2 - test setHours()  3. TC = 3 - test setIsFullTime()  4. TC = 4 - test getEmployee()  5. TC = 5 - test toString()  Enter TC: 3  Enter new isFullTime(True or false): false  OUTPUT:  False | TEST CASE 4  INPUT:  Enter Project ID: 1234  Enter Employee: To An An  Enter IsfullTime (true or false): true  Enter Hours: 120  1. TC = 1 - test getProjectid()  2. TC = 2 - test setHours()  3. TC = 3 - test setIsFullTime()  4. TC = 4 - test getEmployee()  5. TC = 5 - test toString()  Enter TC: 4  OUTPUT:  To An An |
| TEST CASE 5  INPUT:  Enter Project ID: 1234  Enter Employee: To An An  Enter IsfullTime (true or false): true  Enter Hours: 120  1. TC = 1 - test getProjectid()  2. TC = 2 - test setHours()  3. TC = 3 - test setIsFullTime()  4. TC = 4 - test getEmployee()  5. TC = 5 - test toString()  Enter TC: 5  OUTPUT:  1234 To An An true 120 |  |

**Question 2:(3pints)**

Design and code a class named **Product** that holds information about a **Product** andclass named **Food** which is derived from **Product**

Product (code:String, name:String, price: double, quantity: int)

Food (code:String, name:String, price: double, quantity: int ,date:String, expireDate:int)

Food:

Calculate: getPrice()

if getQuantity() <=4, price is original price

if getQuantity()>4 price is discounted by 10 %

The program Menu:

1. TC = 1 - test toString function

2.TC = 2 - test getPrice function

3. TC = 3 - test getName function

|  |
| --- |
| TEST CASE 1  INPUT:  Enter Code: VNP123  Enter name: Canh ga ham  Enter price: 230  Enter quantity: 5  Enter Date: 23/10/2018  Enter expireDate: 1  1. TC = 1 - test toString function  2.TC = 2 - test getPrice function  3. TC = 3 - test getName function  Enter TC: 1  OUTPUT:  VNP123 Canh ga ham 230.0 5  VNP123 Canh ga ham 230.0 5 23/10/2018 1 |
| TEST CASE 2  INPUT:  Enter Code: VNP123  Enter name: Canh ga ham  Enter price: 230  Enter quantity: 5  Enter Date: 23/10/2018  Enter expireDate: 1  1. TC = 1 - test toString function  2.TC = 2 - test getPrice function  3. TC = 3 - test getName function  Enter TC: 2  OUTPUT:  207.0 |
| TEST CASE 3  INPUT:  Enter Code: VNP123  Enter name: Canh ga ham  Enter price: 230  Enter quantity: 5  Enter Date: 23/10/2018  Enter expireDate: 1  1. TC = 1 - test toString function  2.TC = 2 - test getPrice function  3. TC = 3 - test getName function  Enter TC: 3  OUTPUT:  Canh ga ham |

Question 3: (3 points)

Design and code a class named **ClassRoom** that holds information about a ClassRoom

ClassRoom (code:String, name:String, seats:int)

The interface **IClassRoom.** That has 3 methods which were only declared in IclassRoom:

Void f1(**List<ClassRoom> list**)- display all classrooms in the list.

**void f2(List<ClassRoom> list, int fseats, int tseats)** – remove from the list of classrooms "list" all classrooms which has seats from fseats and to tseats.

**int f3(List<ClassRoom> list, String name)** – display all classrooms which are in the list “list” and has name contains given name, and has the number of classrooms below “list”. *The comparison must ignores the case during comparison.*

Design and code a class named **ListClassRoom** which will implement interface IClassRoom and complete 3 methods which were declared in IclassRoom

The program Menu

1. TC = 1 – f1

2. TC = 2 – f2

3. TC = 3 – f3

0. Exit

|  |
| --- |
| TEST CASE 1  1. TC = 1 - f1  2. TC = 2 - f2  3. TC = 3 - f3  0. Exit  Your choice: 1  305L Anfa 305L 50  504L Anfa 504L 31  203 Beta 203 25  203R Anfa 203R 20 |
| TEST CASE 2  Your choice: 2  Enter from seat:24  Enter to seat:30  305L Anfa 305L 50  504L Anfa 504L 31  203R Anfa 203R 20 |
| TEST CASE 3  Your choice: 3  Enter a name: An  3 |

**Question 4:**

You create an interface named **Calculateble**. Design a class **MyCal** which will implement the interface Calculateble:

public String isBeautiful(int number): a number is beautiful which is palindrome and all digits of a number are even.

|  |  |
| --- | --- |
| 123456787654321 86442824468 8006000444422220000222244440006008 235365789787654324567856578654356786556 | Not beautiful  beautiful  beautiful  Not beautiful |

public String separate(String fullname): fullname is separated into first, middle and last name.

fullname: To Thi Anh Duong => first: To, middle: Thi Anh, last: Duong

The program menu:

|  |
| --- |
| TEST CASE 1  1. TC -1 isBeautiful  2. TC -2 separate  0. Exit  Your choice: 1  Enter The number: 1234321  Not beautify |
| TEST CASE 1  1. TC -1 isBeautiful  2. TC -2 separate  0. Exit  Your choice: 1  Enter The number: 22444422  Beautify |
| TEST CASE 2  1. TC -1 isBeautiful  2. TC -2 separate  0. Exit  Your choice: 2  Enter the fullname: To Thi Anh Duong  first: To, middle: Thi Anh, last: Duong |