Assessed Exercise week 2

**Before attempting the tasks, please check the “hints” you can find below each task.**

**Task A**: OOP in C#

Start a Console Application in Visual Studio for C# (.NET framework). Implement a class *Student* that contains name, id and age of the student (assume the name is a string, and id and age are int). Remember to implement the opportune properties (get / set methods) in the class Student.

Also add in the class Student a method *GetInfo()*

public string GetInfo()

{

// to be completed

}

that returns all the info of the student (the method should return a single string containing the concatenation of name, age and id).

In the Main method:

- Create and add 10 objects “Student” in an array of Student objects

- Display the info (name, age, id) for all the students (use a loop) present in the array by using the implemented function *GetInfo*.

*Hints: Start by creating a Console Application for C# in Visual Studio and add a new class Student. Review the material of the Lecture and the Lab in Week 1 on how to start a Console App in Visual Studio and how to implement a class in C# and how to create and read an array of objects.*

*If you are unsure about the syntax of C# consult also the Programming Notes and the suggested resources in the Teaching & Learning Resources on Moodle.*

**Task B:** Implement a Queue of Customers

Create a Windows Forms Application in Visual Studio (GUI) for C# to store and update a *queue* of customer’s names.

The app should allow to store up to 20 customer’s names, and should allow:

1. to add and remove new names to / from the queue (use enqueue / dequeue)
2. display the total number of customers in the queue
3. display the full content of the queue
4. reverse the order of the first k elements of the queue (the user should input k)

*Hints: Follow the material of the Lecture and Lab of Week 2 on the idea of queue, how to implement the queue and how to add a data structure into a GUI (Windows Forms Applications). You can organize the GUI in the way you feel most appropriate and user-friendly (eg using texboxes or listboxes)*

*For d) if the queue is*

*[ “Haniya”, “Sydney”, “Ayub”, “Peggy”, “Tracey”, “Coen”],*

*where “Haniya” is the head of the queue, and k =4,*

*then after applying d) the queue should be*

*[ “Peggy”, “Ayub”, “Sydney”, “Haniya”, “Tracey”, “Coen”]*