Assessed Exercise week 3

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

**Task A: Big O Notation**

Explain in the screencast how to compute the Big O notation for the two segments of code (assume n is the input) so the big O notation should be a function of n.

**1)**

int n = 10;

int ct = 0;

while (n>0)

{

ct = ct + 1;

n--;

}

Console.WriteLine(ct);

**2)**

int n = 50;

int r = 20;

int dummy = 0;

Console.WriteLine("Creating a nested loop");

for (int i = 0; i < n; i++)

{

Console.WriteLine("Inside the first loop");

dummy++;

for (int j = 0; j < n; j++)

{

r = r + dummy;

Console.WriteLine("Inside the second loop " + r);

}

}

*Hints: We have seen examples of how to compute the Big O notation in the lecture. In both cases, assume n is the input so the computed Big O notation must be a function of n. As the example seen in the lecture start the calculation by counting how many times each instruction is executed when the segment of code is run.*

**Task B: Efficient Look-Up of Data**

Dictionary is a data structure where the look-up of data is very efficient.

Implement a Windows Forms Application (GUI) that uses a **Dictionary** and allows to insert, remove and update the availability of an *Employee*.

Each employee has an unique ID (integer) which you can be used as Key of the dictionary. Implement a class *Employee* that has variables to store the *name*, *ID* and *availability* status of the employee (availability is a Boolean variable true / false).

The application should allow the user to:

* Add a new employee (by providing as input the ID and the name of the employee; assume that the availability of a new employee is set to true)
* Remove an employee (by providing as input the ID of the employee)
* Display the info of an employee (name, availability) by providing as input the ID of the employee
* Update the status (true / false ) for the availability of an employee (by providing as input the ID of the employee)
* Display (in a listbox) the IDs of all employees present in the dictionary; allow the user to click on the listbox and select an ID and then show (eg in a separate textbox) the information (name and availability) associated to the employee associated to the selected ID

*Hints: Revise the slides of the lab where we have seen how to use Dictionary in a Windows Forms Application. You can organize the GUI in the way you feel most user-friendly (eg., you can use texboxes to read the input). For the last point, one solution is to use “SelectedItem” for listbox which would allow to detect the item selected in the listbox by the user.*