Assessed Exercise 3

**Task A**

**Calculate the Big O notation for the two segments of C# code shown below**. When you submit your assignment, add in the “Task A” folder a document (eg can be a word or text file) which shows the result of the calculations and explain your calculations in the submitted screencast.

Segment 1:

Console.WriteLine("please input n"); 1 operation

int n = Convert.ToInt32(Console.ReadLine()); 1 operation

int r = 20; 1 operation

int dummy = 0; 1 operation

Console.WriteLine("Creating a nested loop"); 1 operation

for (int i = 0; i < n; i++) 2 operations n times

{

Console.WriteLine("Inside the first loop");1 operation n times

dummy++; 1 operation n times

for (int j = 0; j < n; j++) 2 operations n times

{

r = r + dummy; 1 operations n times

Console.WriteLine("Inside the second loop " + r);1 operations

}

}

Therefore, the O(n^2)

Segment 2:

Console.WriteLine("please input m "); 1 operation

int m = Convert.ToInt32(Console.ReadLine()); 1 operation

for (int j = 1; j <= m; j++) 2 operations n times

{

Console.WriteLine("for loop"); 1 operations n times

}

while (m > 0) 1 operation m times

{

m--; 1 operations m times

Console.WriteLine("while loop"); 1 operations m times

}

Therefore, the O(n)

*Hints to complete the task: We have seen examples of Big O notation calculations in the lecture on complexity (week 7), so please consult the slides on Moodle when attempting this Task.*

**Task B**

Implement an Application in C# (preferably a Windows Forms Application, but a console app with a menu will suffice) that uses a Dictionary and that allows to insert and remove a student, display and update the enrolment status of a student.

Each student has an unique ID (string) which can be used as key of the dictionary. Implement a class *Student* that has variables to store the *name*, *ID* and *enrolment* status of the student (enrolment status is a Boolean variable true or false).

The application should allow the user to:

1. Add a new student (by providing as input the ID and the name of the student); assume that the enrolment status of a new student is set to true
2. Remove a student (by providing as input the ID of the student)
3. Display the enrolment status (true/false) of a student (by providing as input the ID of the student)
4. Update the status (true/false) for the enrolment of a student (by providing as input the ID of the student)

*Hints: Revise the lecture and lab on collections in week 8*

**Task C**

Extend the application implemented in Task B and allow the user to:

1. Display the IDs and names of all students present in the dictionary whose enrolment status is true
2. Display the information (ID, enrolment status) of a student (by providing as input only the name of the student – if multiple students have the same name, display all of them)