CAB201 Programming Principles

First Assignment
Weighting 10%
Due date: 22nd March 2017 at 12pm (noon)

Specification

You are required to write the program which will calculate the **waist to height ratio** of a person. This ratio is a simple measurement used in the assessment of a person's lifestyle risk.

A person's waist measurement is divided by the person's height. For example, a person whose waist measurement is 81cm and whose height of 175cm have a waist to height ratio of 0.463 (rounded to 3 decimal places).

The waist to height ratio works for any unit of measurement as long as both values are the same unit of measurement the ratio is correct. We will assume that the user will enter the values in centimetres. In addition, each value has to be greater than a specified lower limit for each measurement; waist lower limit is 60cm and the height lower limit is 120cm. There is no upper limit on either measurement.

The program will output a person's waist to height ratio as well as informing the person of their risk level of developing obesity related cardiovascular diseases according to the following table:

	Female	**Male**
Low Risk	ratio < 0.492	ratio < 0.536
High Risk	ratio >= 0.492	ratio >=0.536

After the output of the ratio and the risk level, the user will be asked if they wish to perform another calculation of the waist to height ratio. Input of "Y" or "y" will result in another calculation whereas any other input will cause the program to terminate gracefully.

A number of assumptions have been made so that the assignment is able to be done using only the concepts covered in Lectures 1 & 2 and Worksheets 1 - 2. You are free to use any valid C# programming construct, even if it has not been covered in lectures except for the **"goto"** statement. Use of **"goto"** will result in your assignment mark being downgraded substantially by at least 1 grade level.

Your program should consist of one class only containing at least 4 methods in addition to **`Main`** and this count of 4 methods does not include trivial void methods. Examples of trivial void methods are **`ExitProgram()`** and **`Welcome()`** as seen in Lecture 2, slides #21 and #36 respectively

The additional methods should be a mixture of value-returning and void methods. There is no upper limit on the number of methods used. Also each method should have a single purpose.

Assignment Goals

The assignment problem is straightforward. The solution of the problem will be a C# project which will run within Visual Studio IDE.

Though the code could be written as *straight line code*, the sheer number of lines would make the solution difficult to read, understand, and hard to modify as well as containing potential logical errors. **A straight line code solution will not receive full marks even if it fulfils the required functionality.**

The body of the ***Main*** method will resemble a high-level algorithm with minimum number of low level C# constructs. There are be no input or output statements in ***Main***.

Sub-goals of the assignment are to experience:

- Top-down design & development
- Incremental Development & Incremental Implementation
- Translating simple algorithms into C# code
- Writing user defined methods
- Parameter passing
- The mechanics of editing, compiling (building) and running your program
- Testing your program
- Becoming confident, comfortable and competent with programming in the small

Electronic Submission

You will submit your assignment via the link in the Assessment folder on Blackboard (Bb) before 12pm on 22nd March. Detailed information on the assignment submission is available on Bb in the document, **First Assignment Submission Details**

It is recommended that you upload your file to Bb from a lab at QUT. Inability to access Bb from home is not a sufficient reason for submitting the assignment late or requesting an extension.

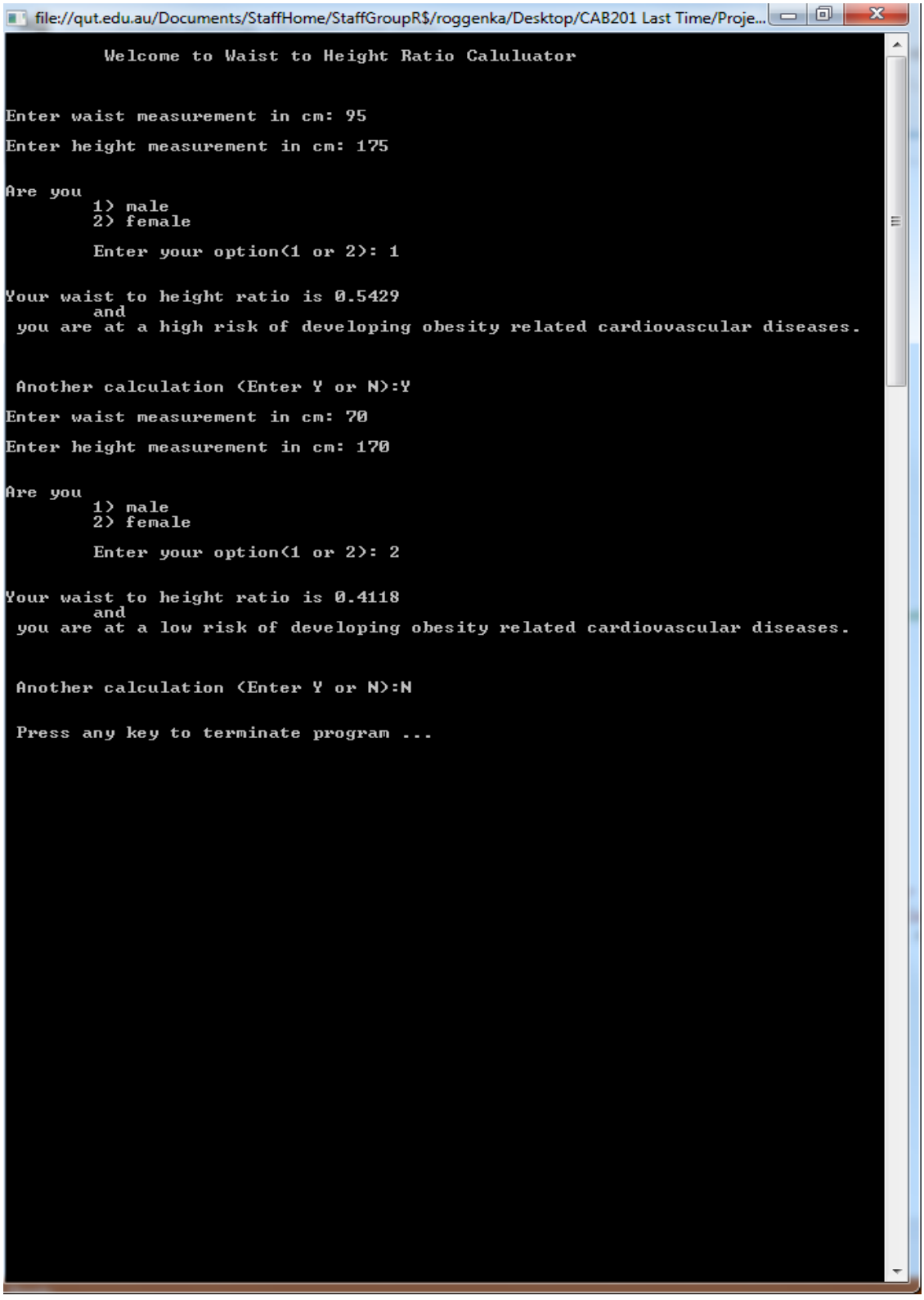
Plan to be finished well before the morning of 22nd March! You should avoid writing all of the code in one sitting and then attempt a build (compilation). Remember ***Incremental Development and Incremental Implementation.***

Final Comments

This assignment is not about screen design so do not waste time attempting to produce the fanciest looking screen output. Though you are free to alter and enrich the output statements it will not gain any additional marks. The basic console interaction on the next two pages is sufficient, as long as it is readable and well laid out on the screen.. There are no marks for “pretty” or ‘clever” screen interactions.

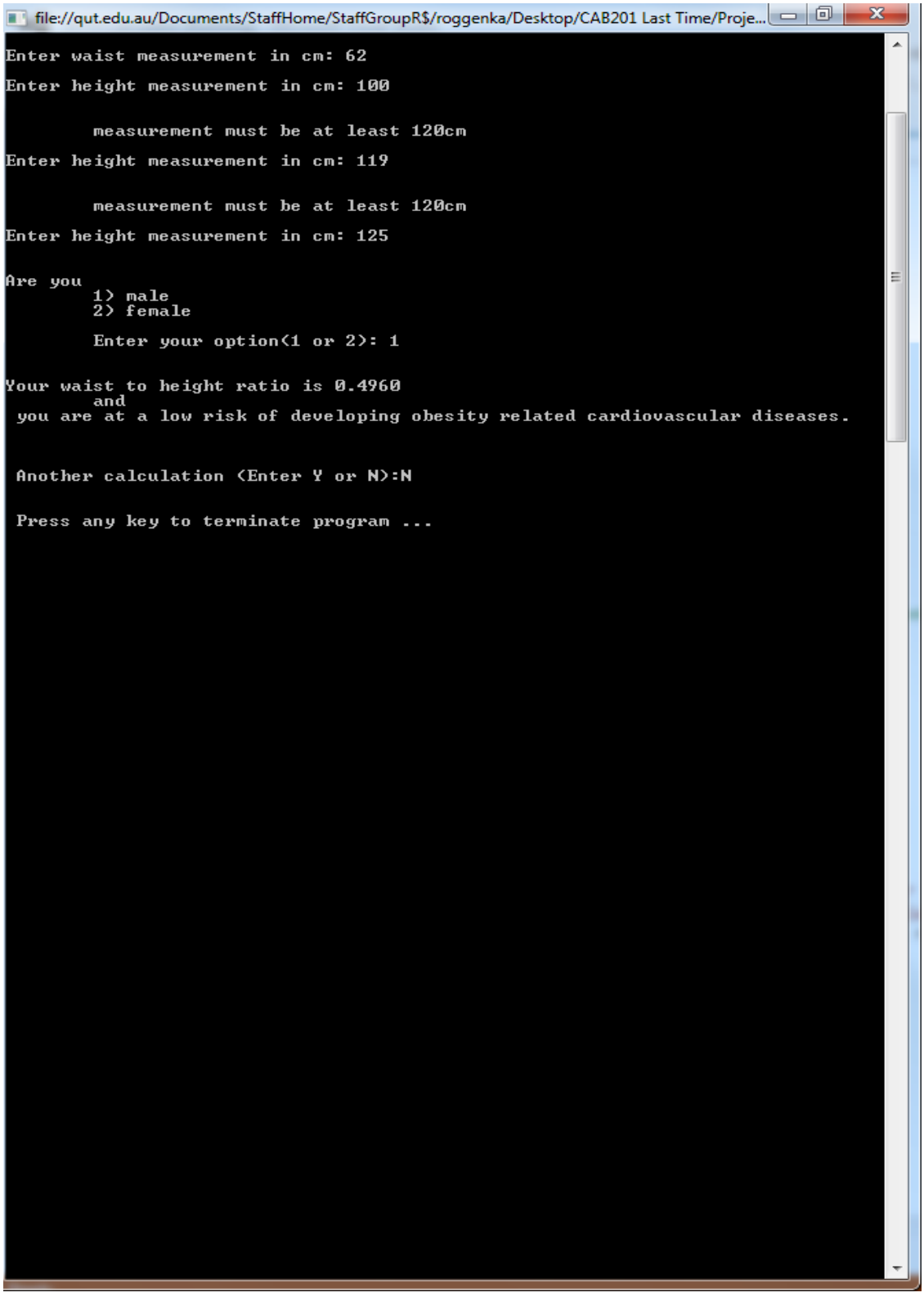
Enjoy the experience of this assignment!

.



.

Screenshot shows simple interaction with valid input values and user wanting to do another calculation and then terminating gracefully.



Screenshot shows detection of values below the minimum specified value for both measurements.

Not all possible interactions have been shown in the two previous screenshots.