**CM3036: Programming in C# - Coursework**

**Hand-out date:** Monday1st February 2016

**Hand-in date:** Thursday 28th April 2016 @10 p.m.

Courseworks that are not submitted by the due time and date will only be accepted if authorisation from the subject lecturer has been given and the coursework is accompanied by a valid explanation.

No coursework will be accepted more than one week after the due date under any circumstances.

**This coursework must be undertaken on an INDIVIDUAL basis. All courseworks may be electronically screened for evidence of plagiarism. Any student who is found guilty of submitting plagiarised work will be awarded a GRADE of F (FAIL).**

**All courseworks must be submitted using the front cover sheet provided for this coursework. The front cover sheet must be SIGNED by the student.**

**The grade for this coursework contributes to the overall assessment of this module in accordance with the module performance descriptor. To pass the module you must obtain an overall grade of D (satisfactory) or better.**

# Introduction

An application is required to collect and save the grade profiles of students who undertake the CM3036 Programming in C# module coursework.

# Requirements

The application should implement functionality that allows the user to:

* collect individual grades for each component of the coursework assessment for each student taking the CM3036 module.
* calculate the overall module grade for each student (as his / her module profile is entered).
* save the individual module grade components and overall grade (persistently in a local SQLServer database).
* perform simple analysis of the saved data.

For the purposes of data collection, the following features and facilities should be implemented:

* The application should prompt the user to enter:
  + student's first name
  + student's last name
  + student's matriculation number
  + grade letter for component 1 (i.e. overall design; weighting factor x 3)
  + grade letter for component 2 (implementation and functionality, weighting factor x 5)
  + grade letter for component 3 (testing, weighting factor x 2))
    - each grade letter will be either A, B, C, D, E or F
* The application should then calculate an overall grade (A, B, C, D, E or F) for each student according to the following table. (N.B. an individual grade is counted a multiple number of times according to its weighting factor).

|  |  |
| --- | --- |
| **Final overall grade** | **Minimal profile** |
| A | at least 5 counted grades @ A;  at least 7 counted grades @ B or better;  all 10 counted grades @ C or better |
| B | at least 5 of counted grades @ B or better;  at least 7 counted grades @ C or better;  all 10 counted grades @ D or better |
| C | at least 5 counted grades @ C or better;  at least 7 counted grades @ D or better; |
| D | at least 5 counted grades @ D or better;  at least 7 counted grades @ E or better; |
| E | at least 7 counted grades @ E or better; |
| F | non-submission or grade profile does not qualify any higher overall grade |

For the purpose of data persistence:

* The application should automatically open access to a local SQLServer database.
* The application should save all data entered for each student (i.e. first name, last name, matriculation number, grade for each assessed component, final overall grade) in a single table of a local SQLServer database.

For the purposes of editing:

* The application should allow the user to edit previously entered results data for the purposes of correcting typographical errors.
* Editing should also include a facility for deleting all results saved for an **individual** student (including the student first name, last name and matriculation number).
* Editing should also include a facility for deleting all data for **all** students currently saved in the SQLServer database.
  + Deletion of all results should only happen after the user has been prompted to confirm his / her intention to remove all data from the database.

For the purposes of data analysis:

* the application should allow the user to generate and inspect a table showing results:
  + the display of results should give a clear visual indication individual students who have passed the module (i.e. achieved an overall final grade of A, B, C or D).
  + the display of results should give a clear visual indication individual students who have failed the module (i.e. achieved an overall final grade of E or F).
  + the display should also display a numerical percentage indicating the proportion of students in the class who have passed the module.

# Constraints

* The coursework solution should be developed as a C# 6 .NET application.
* The C# application must use SQLServer (localdb)\v11.0 as the backing database that saves the user's entered student coursework results. Do not implement any password protection on the database.
* The interaction between the C# application and SQLServer database must be implemented using Entity Framework to provide an Object Relational Mapping (ORM) between your C# code and backing database.
* The user interface must be developed using WPF (Windows Presentation Foundation). (Do not use Windows Forms).
* The application should be developed as a Visual Studio 2013 project. (If you undertake development work using a later version of Visual Studio, please ensure that your project will build and run as a VS2103 project. I will test all coursework submissions using VS2013 only).

# Deliverables

You should submit documentation in hardcopy.

Deliverable D1 (hardcopy):

* brief (and honest) statement of compliance (approximately 1 page long) stating:
  + which requirements have been implemented and verified by testing
  + which requirements have been implemented but not verified by testing
  + which requirements have not been implemented

Deliverable D2 (hardcopy):

* test plan and test results for full application
  + individual test cases should identify
    - purpose of the test
    - expected output
    - actual output

Deliverable D3 (electronic):

* You should submit to a campusmoodle dropbox a single .ZIP file containing:
  + your project report, as .docx file
  + a complete VS2013 project folder containing full source code for your coursework solution.
  + SQLServer .mdf and .ldf files for the localdb\v11.0 database must also be submitted
* The name of the zip file should be composed from a combination of your name and matriculation number e.g. *ddavidson123456.zip*.

# Assessment

|  |  |
| --- | --- |
| **Assessed Element** | **Weighting** |
| Overall design | x 3 |
| Implementation and functionality | x 5 |
| Testing | x 2 |

The final overall grade for the module is based on the profile of weighted grades awarded for each of the assessed elements. An element weighted “x 2” carries a grade that is counted twice in the overall grade profile, an element weighted “x 3” is counted three times in the overall profile, an element weighted “x 5” is counted five times in the overall profile etc.

|  |  |
| --- | --- |
| **Final overall grade** | **Minimal profile** |
| A | at least 5 counted grades @ A;  at least 7 counted grades @ B or better;  all 10 counted grades @ C or better |
| B | at least 5 of counted grades @ B or better;  at least 7 counted grades @ C or better;  all 10 counted grades @ D or better |
| C | at least 5 counted grades @ C or better;  at least 7 counted grades @ D or better; |
| D | at least 5 counted grades @ D or better;  at least 7 counted grades @ E or better; |
| E | at least 7 counted grades @ E or better; |
| F | non-submission or grade profile does not qualify any higher overall grade |