# Introduction

**CS6004ES –Application Development Individual Coursework –Main Sit(2021/22)**

This individual coursework requires developing and documenting a small Students marks tracking Window based application in C# using an object oriented approach and Visual Studio IDE.

Your application must provide abilities such as registration of students, staff, subjects in classes, documenting of grades and analytical marks of each student and other evaluation elements. In addition, an SMS is used for planning student’s curriculums, and managing all student-related needs within a school

Your software artefact must be submitted as a Visual Studio project. It will be assessed using Visual Studio 2015 or any higher version and any features not working in the standard installation of Visual Studio 2015 or any higher version will not be assessed.

The coursework carries 30% of the module mark.

**Set:**

**Submission Deadlines:23rd July 2021**

**Coursework Submission in-class Demo: 25th July 2021**

This individual coursework has 2 parts, both of the soft copies which are to be submitted before 11.59 pm on

**23rd July 2021**

1. **The software application to be developed in C#**
2. **The documentation in MS Word format.**

*Please note the rules on plagiarism*

The application should be implemented individually. This is not a group/team effort. Any material which is a direct copy from someone else (student or other source) or a close paraphrase/code must be indicated where it is quoted i.e. it must be made clear what material is a quotation or close paraphrase e.g. by showing the text in italics or in quotation marks. It is not sufficient to show the source in a list of references or bibliography. If you are unclear, please discuss your examples with your seminar tutor or the module leader. Plagiarism is a serious offence and conviction for plagiarism may lead to suspension from the University, even for a first offence (please see the section on Academic Misconduct in the Student Handbook).

**Case study**

OMP International School stands out from the crowd as one of the Top schools in Sri Lanka. They employ the strategy of imparting knowledge with the use of both modern and indigenous methodologies. Due to their unique approach to education, they have been honoured with several awards and rank among the best schools in Sri Lanka.

OMP International School is trying to create a Student Management and Marks tracking software (e-pupil) that can manage students and their marks details. This software further makes it easier for officials to finish off their work in a lesser span of time.

“e-pupil” system is aimed at total user-friendly as well as efficient management of varied tasks. These tasks may range from registering new students, staff, subjects, mapping subjects to classes, mapping students to classes and examination details for making the administrative division of school effective. In order to cope up with all these factors, the school management system needs to be developed.

IT department of OMP International School is decided C# as the target language and 2 Roles (Admin and Staff) in “e-pupil”.

Required tasks are listed down below.

Admin

1. Only Admin can manage school details (make sure to have at least 5 attributes).
2. Only Admin can manage class details (make sure to have at least 3 attributes).
3. Only Admin can manage student details (make sure to have at least 6 attributes).
4. Only Admin can manage staff details (make sure to have at least 8 attributes).
5. Only Admin can manage subject details (make sure to have at least 5 attributes).

Staff

1. Only staff can map subjects to classes.
2. Only staff can map students to classes.
3. Only staff can manage marks details of each student.
4. Only staff can generate various reports (make sure to have at least 3 reports).

***Note****: You may add extra features - both data and functionality to the application, if you wish.*

***Your software implementation should demonstrate/provide the following features***

1. Use of appropriate data types (built-in and programmer-defined) to handle the application data
2. Use of appropriate data structures for the required programming scenario
3. Use suitable algorithms e.g. sorting
4. Define and use your own class or classes
5. Provide window-based user interface for your application
6. Store the data related to the application

**Deliverables**

Your submission should include the software project and a reflective essay as described below.

1. Your software artefact in the form of a Visual Studio 2015 project, which should include the program’s source code, compiled classes, the executable file and data file (if any).
2. A reflective essay (1000 or more words), which concisely documents:
   1. Detailed instructions to run the program
   2. The architecture of your software in terms of software classes, clearly indicating which classes to be of your own work and which classes from other sources (e.g. From textbooks, online sources such as MSDN etc.).
   3. Detailed description of the classes’ properties and methods
   4. A description of your algorithm to calculate the total income per any given month and sort the customer by loyalty in the forms of flowcharts and/or decision tables.
   5. Which data structures and which algorithms you have used, in which part of your program, and why.
   6. Your reflection of own experience of using C# and visual studio for the development task, which feature you like and why, what issues you experienced and your solution to overcome it.

**Marking Scheme for the CS6004ESIndividual Coursework**

This individual coursework counts for 30% of the module mark. The following are guidelines for marking. Mark each item listed below on a scale 0 to 5 where the marks correspond. Then multiply the mark by the weighting indicated, total and divide by 2 to get the total mark.

|  |  |
| --- | --- |
| Mark | Characterised by |
| 0 | No work or work totally irrelevant |
| 1 | Work started on right lines but no result |
| 2 | Some result, with major lack and/or errors |
| 3 | Acceptable result but incomplete, or some good result with minor errors |
| 4 | Good result but can be further polished |
| 5 | Excellent result |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Item** | **We ight** | ***Mar to 5)*** | ***Weight x Mark*** |
| **Implementation** | |  |  |  |
| 1 | The application user interface | 3 | 5 | 20 |
| 2 | Task1: Only Admin can manage school details | 2 | 5 | 10 |
| 3 | Task2: Only Admin can manage class details | 2 | 5 | 10 |
| 4 | Task3: Only Admin can manage student details | 3 | 5 | 10 |
| 5 | Task4: Only Admin can manage staff details. | 2 | 5 | 10 |
| 6 | Task5: Only Admin can manage subject details. | 3 | 5 | 10 |
| 7 | Task6: Only staff can map subjects to classes | 3 | 5 | 10 |
| 8 | Task7: Only staff can map students to classes. | 2 | 5 | 10 |
| 9 | Task8: Only staff can manage marks details of each student | 2 | 5 | 10 |
| 10 | Task9: Only staff can generate various reports | 2 | 5 | 10 |
| 11 | Task10: Staff Dashboard and Admin Dashboard |  |  | 10 |
| **Documentation** | |  |  |  |
| 1 | Detailed instructions to run the program | 1 | 5 | 5 |
| 2 | The software architecture | 2 | 5 | 10 |
| 3 | Description of your algorithm to sort the customers according to their loyalty | 2 | 5 | 10 |
| 4 | Detailed description of the classes’ properties and methods | 2 | 5 | 10 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | Data structures and which algorithms used in the project | 2 | 5 | 10 |
| 6 | Own reflection of own experience | 2 | 5 | 10 |
| **Programming style** | |  |  |  |
| 1 | Clarity of code which shows the underlying algorithm | 1 | 5 | 5 |
| 2 | Sensible naming of programmer-defined variables, classes, properties and methods | 1 | 5 | 5 |
| 3 | Useful comments in code | 1 | 5 | 5 |
| 4 | Data validation and exception handling | 1 | 5 | 5 |
| 5 | Interface design and usability of the system | 1 | 5 | 5 |
|  | **Total** |  |  | **200/2** |