ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 1: Programming		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name		**Student ID**	
Class		**Assessor name**	Hong-Quan Do
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

Grading grid

P1	M1	D1

❒ **Summative Feedback:**		❒ **Resubmission Feedback:**
Grade:	**Assessor Signature:**	**Date:**
Lecturer Signature:		

Assignment Brief and Guidance:

Scenario: You have applied for a post as a trainee with a software development company and have been invited for an interview. You have been asked to demonstrate your problem solving and basic programming skills. To do this you have to prepare a report on using algorithms to solve problems.

You need to explain, using examples, how algorithms are used to solve simple business problems and the steps needed to be followed to produce a working program solution. You should make clear your assumption about your program. The problems to be solved will involve basic procedural programming instructions - sequence instructions (input, output and assignment statements), loops, and conditional statements. Problems should be analyzed and designed by the use of flowchart and demonstrated by the use of modules (procedures) using a **menu-based program**.

Tasks:

1. State your simple business problems to be solved.
2. Analyze the problem and design the solutions by the use of suitable methods.
3. Demonstrate the compilation and running of a menu-based program
4. Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language.

You also need to do a presentation of your work (it should be summary of your report).

HERE MY GUIDANCE

- In order to achieve P1: you are recommended to process task 1.2, 2.1 (Use case diagram is must-have), task 2.2 (flow charts are must-have, at least 2 diagrams), task 3 (provide screenshots of your own project), task 4.1 and 4.2.
- In order to achieve M1: P1 + good task 3.
- In order to achieve D1: P1 + M1 + good task 4 (including 4.3)

1. STATE A SIMPLE BUSINESS PROBLEMS TO BE SOLVED. (2 pages)

1.1. Introduction

- I have recently applied for a post as a trainee with a software development company. As being prepared for an interview, I am going to demonstrate problem solving and basic programming skills that I have learnt from the course Programming. (Paraphrase !!!)

1.2. Define the problem

E.g 1.

A math teacher wants to manage grades of a class. He asks you to help him to write a small application to do that. He needs to enter student IDs, student's grades and store these information into 2 separate arrays (integer array for IDs and float array for grades). Then he needs to print all student IDs together with their grades. Finally, he needs to know which student has highest grade and lowest grade. Your program should be menu based with the options above. When an option is done, the program should go back to the main menu so he can choose another option. There should be an option to quit program.

Other examples:

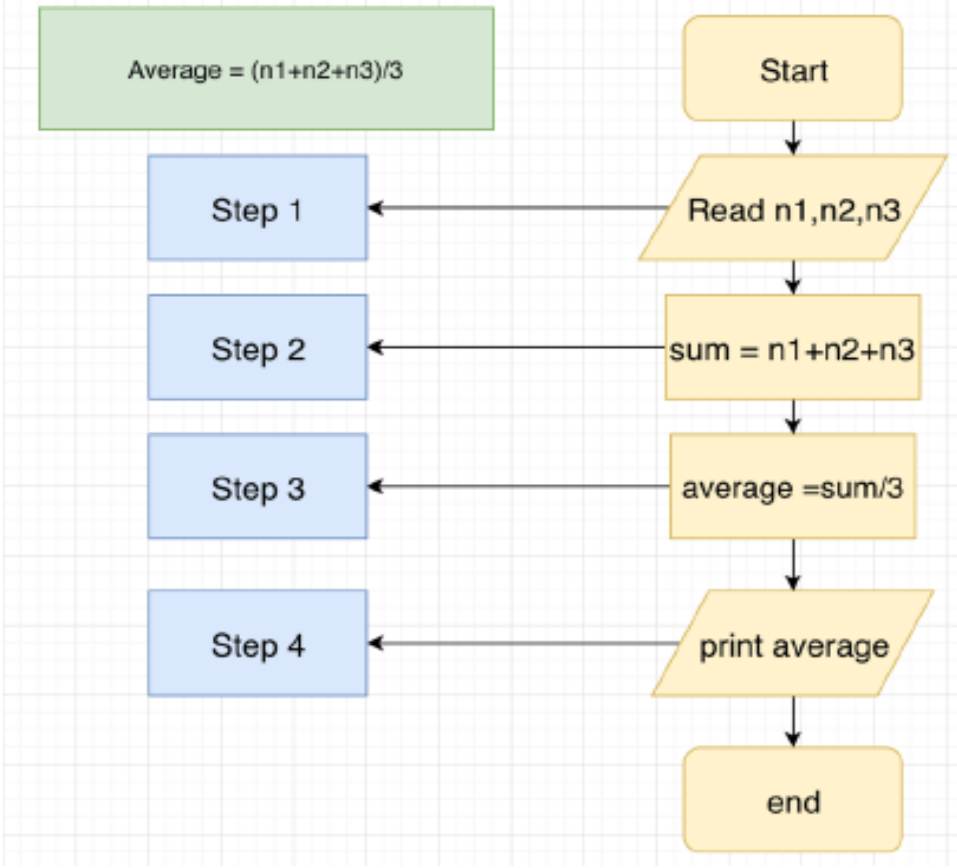
- The problem that manages the marks's students.
  - Assign marks to students
  - Calculate the average marks (mean value), median value
  - Count how many students whose'marks greater/less than the average marks
- The problem of how to manage and evaluate ratings for movies

- The problem of managing books in library
  - Add book(s)
  - Search book(s)
  - Print all books (by category

2. ANALYSE THE PROBLEM AND DESIGN THE SOLUTIONS BY THE USE OF SUITABLE METHODS. (5-10 pages)

2.1. Analyze the problem

- Define what is an algorithm (Slide - lesson 1), how they can help in problem solving.
  - Nếu ra định nghĩa chung nhất, có thể dẫn nguồn
  - Mô tả bằng hình ảnh. Ví dụ:



- Steps in Program Development

Figure 1. *SDLC Process (Source: Agarwal M., techbeamers.com, 2020)*

The various steps involved are:

- Defining or Analyzing the problem
- Design (Algorithm)
- Coding
- Documenting the program
- Compiling and running the program
- Testing and Debugging
- Maintenance

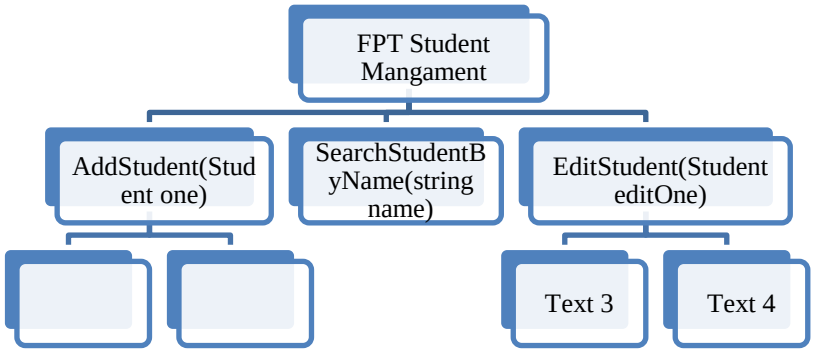
In each step, you need providing some discussion of what it is about.

- Use-case diagram should be designed and provided here for a better understanding of the requirements.
- Details of the problem should be stated.

2.2. DESIGN THE SOLUTIONS

- Clarify components, data, structures needed to implement the given solution.
  - Programming language that you use, Supporting tools, Components in C# Project
  - Data you use
    - Data types in C#, Variables should be used.
  - Structures you use
    - Conditional statements
    - Iteration statements
- Procedural programming paradigm: some key characteristics, focus on explaining how the use of functions

  Present a diagram showing the way that the main program can be divided into sub-functions



- Pseudo code/flow chart for main functionalities of the program

3. DEMONSTRATE THE COMPILATION AND RUNNING OF A MENU-BASED PROGRAM (1-2 pages)

- Explain the way the source code is compiled and run (Example: Figure 2 OR Figure 3)
    - https://codeasy.net/lesson/c_sharp_compilation_process
    - https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework
- In each step, provide screenshots of your own project (results of compilation and running) in Visual Studio/SharpDevelop as evidence. (Example: Figure 4)

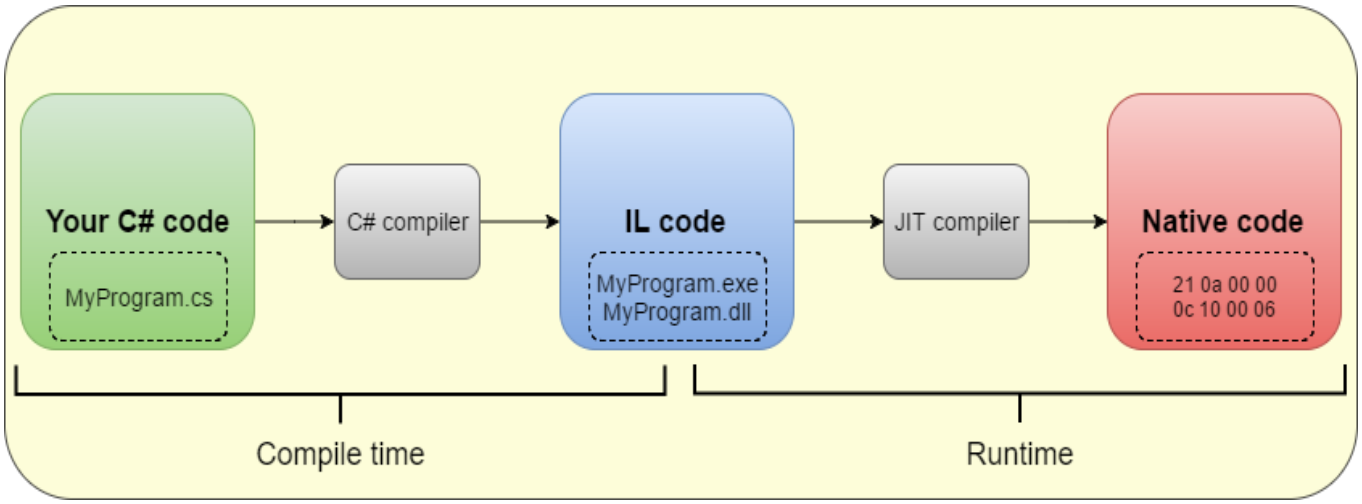


Figure 2. *Compilation and running of a C# program (Source: codeasy.net, 2020)*

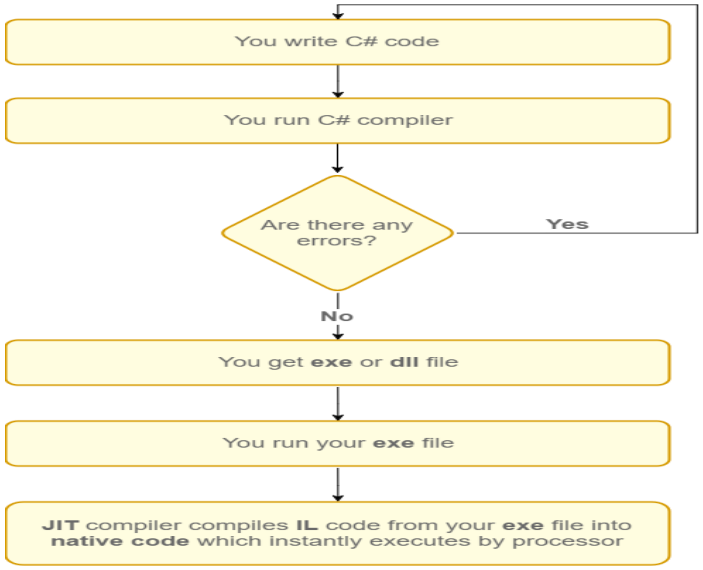


Figure 3. *Flow chart of how to compile and run a C# program (Source: codeasy.net, 2020)*

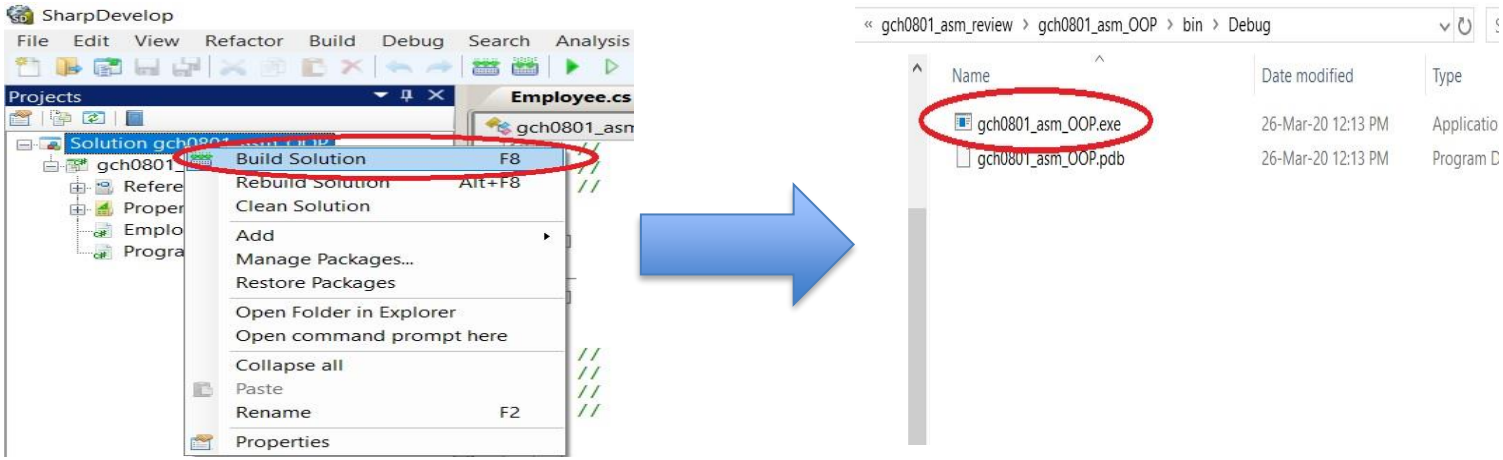


Figure 4. *The compilation of my project*

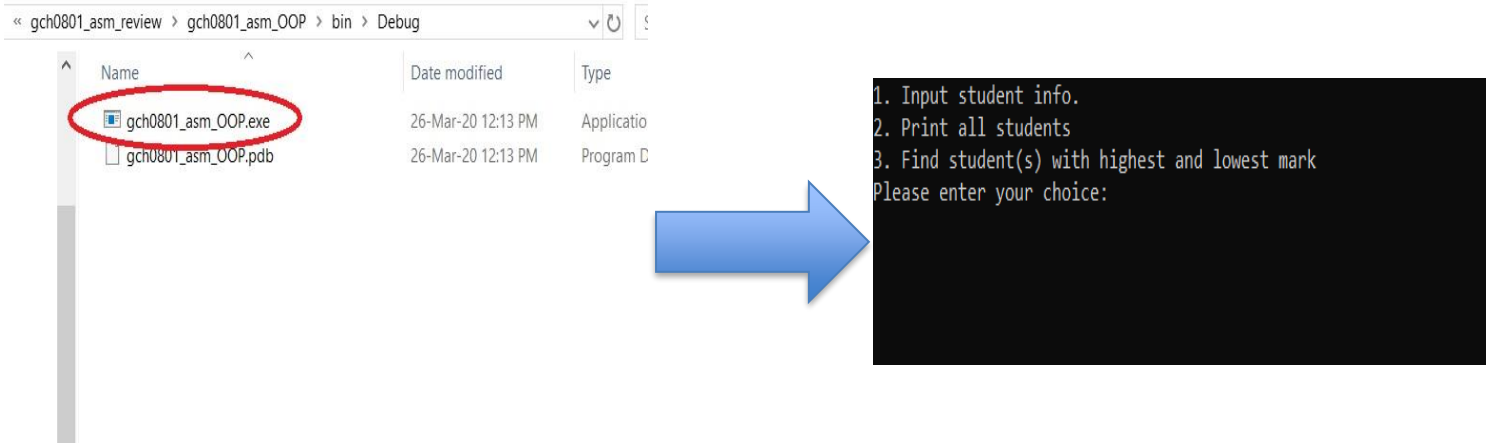


Figure 5. *The running of my project*

4. EVALUATE HOW THE PROBLEM IS SOLVED FROM THE DESIGNED ALGORITHM TO THE EXECUTION PROGRAM WRITTEN BY A SPECIFIC PROGRAMMING LANGUAGE. (5-10 pages)

4.1.Show how the program works: Provide snapshots of the program when it runs.

4.2.Explain the way implementing main functionalities

Provide the source code of each main functionality, explain the implementation.

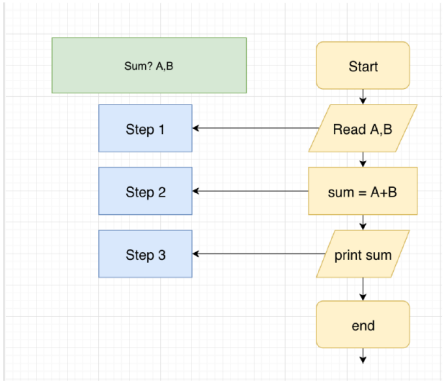
4.3.Evaluation

- Pros/ Cons/ Suggestions for improvement
- Relationship between the written algorithm and the code variant
  - An algorithm is defined as a well-defined sequence of steps that provides a solution for a given problem. It doesn't specify how the solution is build -> Encapsulate implementation details
  - The programmer takes many other things into consideration like memory management, data types, data manipulation etc., alongside with algorithms and data structures, in order to write the code variant -> More complex than the "written algorithm".

Evidence: comparison of your own algorithm with code variant (compare it with the design of flow chart that you provided). Example:

Example: Sum of two numbers A and B

- Alogorithm:



Program ▸ Main(string[] args)

```
using System;

namespace sum
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            int A = 3, B = 5,sum ;
            sum = A + B;
            Console.WriteLine("Sum of two numbers A and B:{0}", sum);
            Console.Read();
        }
    }
}
```

REFERENCES

(Apply Harvard referencing style)

- **Book Referencing Example:**
  Mitchell, J.A. and Thomson, M. (2017) *A guide to citation*.3rd edn. London: London Publishings.

- **Journal Article Example**
  Mitchell, J.A. 'How citation changed the research world', *The Mendeley*, 62(9), p70-81.

- **Journal Article Online Example**
  Mitchell, J.A. 'How citation changed the research world', *The Mendeley*, 62(9) [online]. Available at: https://www.mendeley.com/reference-management/reference-manager (Accessed: 15 November 2016)

- **Website Example:**
  Mitchell, J.A. (2017) *How and when to reference* [Online]. Available at: https://www.howandwhentoreference.com/ (Accessed: 27 May 2017)