CPT106 C++ Programming and Software Engineering II

# Individual Project

## Fundamental of C++ language

| | |
|---|---|
| Project Number | 1 |
| Contribution to Overall Marks | 20% |
| Release date | 15 March 2023 |
| Submission Deadline | Tuesday, 20 April 2023, 23:59 |

## How should the work be submitted?

***SOFT COPY ONLY !***

(MUST be submitted through Learning Mall so that we can run your programs during marking.)

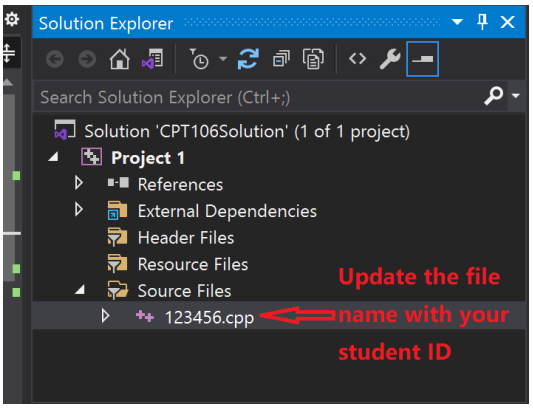
## Assessment Overview

This assessment aims at testing some basic concepts of C++ programming and coding in the software development environment (**Visual Studio**). To distribute the workload of the individual project more evenly throughout the whole semester, we divided it into 3 smaller individual projects for you to complete separately. This is the first individual project.

## What should be submitted?

Only your C++ source code file (,cpp) should be submitted to Learning Mall (LM). Your grade will also be given through LM. There are several steps for you to note：

1. You should create a solution named “CPT106Solution”.
2. Create a project named “Project 1”.
3. Create one CPP source file of “123456.cpp” (123456 is your student ID) in Project 1.
4. All the classes and the main function should be in the single source code file (“123456.cpp”).
5. Include clear comments in your code to make it easy to understand.
6. **Only submit the cpp file to the LMO**. For example, “123456.cpp” in “Project 1” should be submitted.
7. ALL programs MUST compile and run in Visual Studio 2013/2016/2019/2022.
8. Your project structure may look like this:


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'CPT106Solution' (1 of 1 project)
Project 1
References
External Dependencies
Header Files
Resource Files
Source Files
123456.cpp
Update the file name with your student ID

**Project 1 (100)**

Write a C++ program that can perform the following:

1. Define a class **NamePyramid** in your program with the following functions. **(20 marks)**
   a. Define a function **void namePyramid_under(string name)** to print a pyramid with * character followed by your name at the bottom of the pyramid (10 marks)

      **For example:**

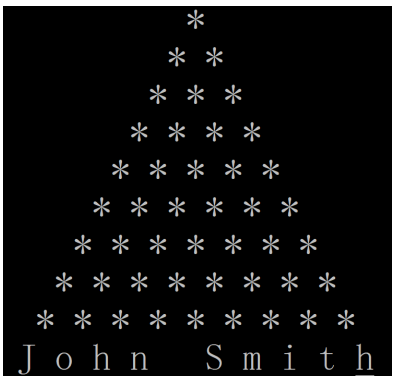
      **Input name:** Jack Ma

      **Output:**

      **Input name:** John Smith

      **Output:**

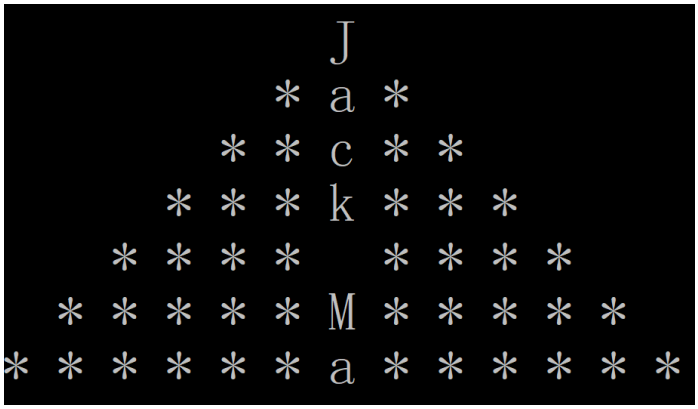
      
      

   b. Define a function void namePyramid_mid(string name) to print your name vertically at the middle of the pyramid shape. (10 marks)

      **For example:**

      **Name:** Jack Ma

      **Output:**

2. Define a class **ArrayOperation** in your program with the following functions. **(40 marks)**
   a. Define a function **void ArrayInput ( int a[], int lenA)** to init an integer array. (10 marks)

      Requirements:

      1) The array is inputted from the keyboard.
      2) The array has ***lenA*** elements.

   b. Define a function **int maxElement (int a[], int lenA)** to return the maximal value of the element in the array. (10 marks)

   c. Define a function **void maxArray(int a[], int lenA, int b[], int lenB)** to compare the sum of element values of **array a to array b**. If the sum of **array a** is larger than array b, print out a string "the sum of array A is larger than the sum of array B". Otherwise,

print out a string "the sum of array A is smaller than or equal to the sum of array B". (10 marks)

d. Define a function **int NumofSameElements (int a[], int lenA, int b[], int lenB)** to check and return the number of the same elements in two arrays, ignoring the order and multiplicities. For example, the number of the same elements in the two arrays {1, 4, 16, 11, 7, 9, 11} and {11, 4, 7, 9, 16, 4, 13, 100} is 5 (they are 4, 16, 11, 7 and 9). The number of the same elements in the two arrays {3,4,5} and {6, 7, 8} is 0. (10 marks)

3. Define a main function **void main()** in your program to test the functions in the previous two questions. Within the main function, you should do as follows: **(10 marks)**

   Requirements:

   a. Create an object of NamePyramid, input your name by keyboard, and output the two name pyramids with your name correctly.

   b. Create an object of ArrayOperation to complete the following functions:

      1) Define two arrays in the main function using two constant variables as the array length.

      2) Init the two arrays using the function **ArrayInput** in **ArrayOperation** class.

      3) Call the function **maxElement** and print out the maximal value of the two arrays, respectively.

      4) Call the function **maxArray** to print out the comparison result.

      5) Call the function **NumofSameElements** and print out the number of the same elements in two arrays.

# Marking Scheme

| | |
|---|---|
| Function definition | 70% (20%,40%,10%) |
| Comments | 10% |
| Correction & Robustness | 20% |