CPT106 C++ Programming and Software Engineering II

Assessment 1

Fundamental of C++ language

Assessment Number	1
Contribution to Overall Marks	15%
Submission Deadline	Tuesday, 20 April 2021, 23:59

How the work should be submitted?

SOFT COPY ONLY !

(MUST be submitted through Learning Mall so that we can run your programs during marking.) Make sure your name and ID are printed on the cover page of your report.

Assessment Overview

This assessment aims at testing some basic concepts of C++ programming and initiates the routine of code development using the software development process (**SDP**), namely the five main steps of the software development process:

1. Problem statement: formulate the problem.
2. Analysis: determine the inputs, outputs, variables, etc.
3. Design: define the list of steps (the algorithm) needed to solve the problem.
4. Implementation: the C++ code has to be submitted as a separate file. Just indicate here the name of the file.
5. Testing: explain how you have tested and verified your C++ program.

You will need to apply this methodology to each one of the following simple exercises.

What should be submitted?

A short ***report*** (up to a few pages of texts plus C++ source codes) detailing for all the questions of the assignment. The answer for each question should follow the SDP method:

a) Overall quality of report (10% of the total marks for that question)
b) SDP steps 1 to 3. (30%)
c) SDP step 4 (implementation): your C++ source code including the comments. (40%)
d) SDP step 5 (testing): you will explain how you have tested the correctness of your C++ program and will include some sample runs of your C++ Programs. (20%). **The testing result must be shown by screenshot.**

The report in either Microsoft Word format **(.DOCX file)** or **PDF format** together with **C++ source code** for all questions should be zipped into ***a single file***. (For maintenance purposes, it is always a good practice to comment your code as you go. The comments state the aim of the program, what are the inputs, what are the outputs, which algorithm is used, who is the author and so on.)

EXERCISE 1 (5 POINTS OUT OF 15)

Write a function

```
bool same_Array (int a[], int lenA, int b[], int lenB)
```

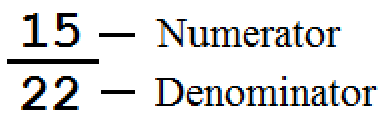
that checks whether two integer arrays have the same elements, ignoring the order and multiplicities. For example, the two arrays {1, 4, 16, 11, 7, 9, 1, 11} and {11, 4, 7, 9, 16, 4, 1} would be considered identical.

Requirements:

1. arrays are inputted from keyboard.
2. The length of two arrays should be different.
3. Call the function same_Array in your main function. Initialize two arrays. Print out “same” if the function returns true. Print out “different” if the function returns false.

EXERCISE 2 (10 POINTS OUT OF 15)

Design a new class to represent a fraction (a ration of two integer values).



The data members of the class **Fraction** are two integers, top and bottom, denoting the numerator and denominator, respectively.

```
class Fraction
{
private:
    int top;         // Numerator
    int bottom;      // Denominator
public:
    . . . . . .
};
```

Part 1: Fundamental requirements for the class Fraction. (5 POINTS)

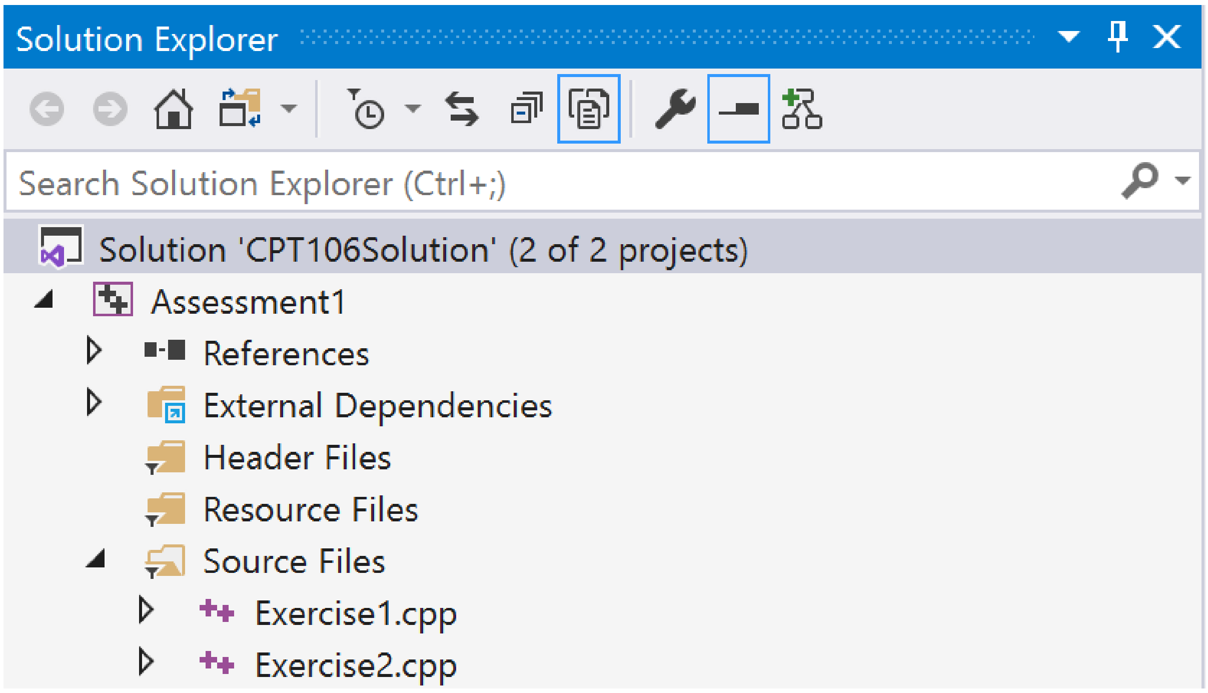
1. Fractional numbers can be declared with both a numerator and denominator or simple numerator:
   ```
   Fraction a;             // represents 0/1
   Fraction  b(3,4);  //  represents  3/4
   Fraction c(5); // represents 5/1
   ```
2. Fractions should be able to act just like other numbers.
   Add, subtract, multiple and divide;
   Compare based on values;
   Input and output.

Part 2: Advanced requirements **(5 POINTS)**

3. The fractional number is normalized to ensure that only the numerator can be negative and the value is in the least common denominator form:
   2/-3 would be converted into -2/3 automatically;
   15/21 would be converted into 5/7 automatically.
4. Write methods to convert between decimals and fractions.
5. Create a Fraction object and test these methods above in your main function.

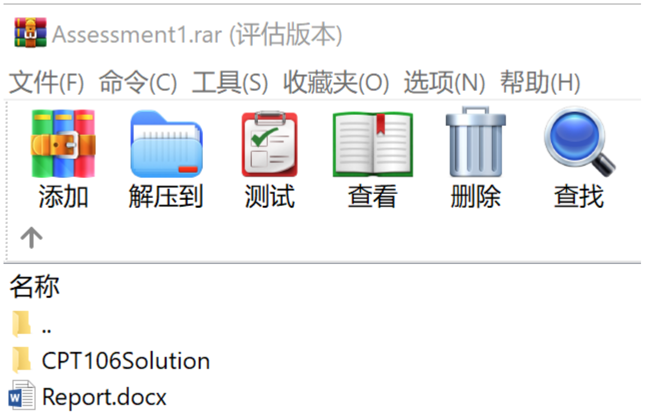
The recommended submission step and structure of Assessment 1

1. You should create a solution named “CPT106Solution”.
2. Create a project named “Assessment1”.
3. Create two CPP source files of “Exercise1” and “Exercise2” respectively in the project “Assessment1”.



```
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'CPT106Solution' (2 of 2 projects)
  Assessment1
    References
    External Dependencies
    Header Files
    Resource Files
    Source Files
      Exercise1.cpp
      Exercise2.cpp
```

4. Write your code in these two CPP files, respectively.
5. Because one project only can have one main method. So you can comment out the main method in one CPP source file when you run another CPP source file.
6. ZIP the whole C++ project and your report.



Assessment1.rar (评估版本)

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找

```
名称
..
CPT106Solution
Report.docx
```

7. Upload the ZIP file (Assessment1) to Learning Mall.