SEHH2042 Computer Programming

Individual Assignment 1
Submission deadline: 5-Mar-2020 (Fri), 18:00

Expected Learning Outcomes

- Familiarise themselves with at least one high level language programming environment.
- Develop a structured and documented computer program.
- Apply the computer programming techniques to solve practical problems.

Introduction

This is an **individual assignment**. You are given a C++ program template file called *A1Template.cpp*. You are required to **insert C++ codes into the template file** according to the given instructions. The final program should be able to satisfy all the requirements in this specification.

Instruction

- To answer the questions, you need to insert codes into the functions *as specified in the template file*. E.g., to answer question 1, write your code in the scope of *Q1( )*. When the program is executed, enter the question number to run the code of a particular question.
- You may write user-defined functions to solve the questions. E.g., you may write a user-defined function for solving question 1, and call it in the given function *Q1( )*.
- You can include more header files in the template file if necessary. ***(refer to appendix)***
- Apart from inserting codes as mentioned above, you are **NOT** allowed to modify any given codes in the template file.
- You may assume that user always provides valid input. **NO error checking** is required unless required by the question.
- You need to do **either odd or even version** of a question according to your student ID number, and follow **EXACTLY** the requirement and sample output format as stated in the questions. **NO extra display text** (e.g. please enter…) is required.
- **IMPORTANT**: Make sure that your file **CAN be opened**, and has **NO syntax error**.

ShowInfo

To show your identity, insert your personal particulars as stated in the *showInfo* function.

Sample display:

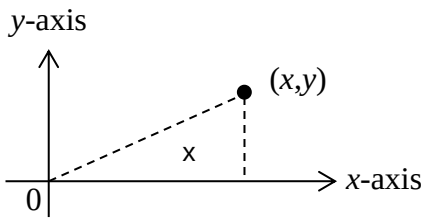
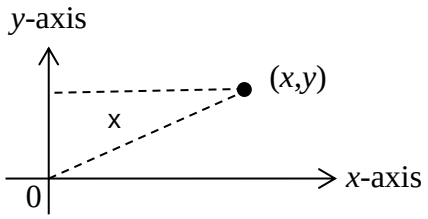
```
Name      : XXX YYY ZZZ
Student ID: 20xxxxxxA
Class     : 201A
```

Question 1 (25%)

Write your code in *Q1( )* that determines the centroid of a right-angled triangle on a (*x*,*y*)-coordinate plane. Given the coordinates of three vertices (x_1,y_1), (x_2,y_2) and (x_3,y_3), the centroid is calculated as

$$\left(x_c = \frac{x_1 + x_2 + x_3}{3},\ y_c = \frac{y_1 + y_2 + y_3}{3}\right)$$

The user inputs the coordinate of a vertex (*x*,*y*), that forms a right-angled triangle with the origin and *x*-axis (or *y*-axis). Then your code should display the coordinate of the centroid as shown in the sample display. Use **double data type** for all numeric values. Display the calculation result in **2 decimal places**.

3rd digit of Student ID	Use axis	Centroid (x)
Odd	*x*-axis	
Even	*y*-axis	

Sample display (for odd)

```
Input x: 123.456
Input y: 12.34
Centroid is at (82.30,4.11)
```

Sample display (for even)

```
Input x: 123.456
Input y: 12.34
Centroid is at (41.15,8.23)
```

Note: You need to follow all text and the input order (input *x* first) as shown in the sample display. Be careful to the spacing and spelling mistakes.

Question 2 (25%)

Write your code in *Q2( )* to solve a quadratic equation of the form $ax^2 + bx + c = 0$. Your program accepts three numbers, *a*, *b*, and *c* **(double data type)**, and prints the solution(s), if any, of the equation in default format (i.e. 6 meaningful digits). You may assume that *a* is non-zero.

There may be <u>two, one or no solution</u> for a quadratic equation, which can be determined by the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

If there are two distinct solutions, **<u>display them in ascending order</u>** in a single line separated by a space (see sample display 1). If the equation has no solution, print "No solution" (see sample display 3).

<u>Sample display 1</u>
```
Enter a b c: 1 3.5 2
-2.78078 -0.719224
```

<u>Sample display 2</u>
```
Enter a b c: 1 2 1
-1
```

<u>Sample display 3</u>
```
Enter a b c: 10 20 30
No solution
```

Hint: To calculate square root of a number, you need to include header file <*cmath*> and use the function *sqrt( )*.

Example:
```
double x = 151.29;
double ans = sqrt(x);  // ans stores 12.3
```

<u>Question 3 (25%)</u>

Write your code in *Q3( )* that calculates how much money you will have after each day of spending (or saving). The user inputs the starting (or target) amount of money (**integer data type**). Then In each day, the amount of money you spend (or save) is the same as the day number. The program lists out the spending (or saving) progress from day 0 until using up all the money (or reaching the saving target).

5th digit of Student ID	**Input $**	**Calculate**
Odd	Original amount	Spending
Even	Saving target	Saving

<u>Sample display (for odd)</u>
```
Starting money: $6
Day 0 you spend $0 so you have $6
Day 1 you spend $1 so you have $5
Day 2 you spend $2 so you have $3
Day 3 you spend $3 so you have $0
```

<u>Sample display (for even)</u>
```
Target money: $6
Day 0 you save $0 so you have $0
Day 1 you save $1 so you have $1
Day 2 you save $2 so you have $3
Day 3 you save $3 so you have $6
```

<u>Sample display (for odd)</u>
```
Starting money: $20
Day 0 you spend $0 so you have $20
Day 1 you spend $1 so you have $19
Day 2 you spend $2 so you have $17
Day 3 you spend $3 so you have $14
Day 4 you spend $4 so you have $10
Day 5 you spend $5 so you have $5
Day 6 you spend $6 so you have $-1
```

<u>Sample display (for even)</u>
```
Target money: $20
Day 0 you save $0 so you have $0
Day 1 you save $1 so you have $1
Day 2 you save $2 so you have $3
Day 3 you save $3 so you have $6
Day 4 you save $4 so you have $10
Day 5 you save $5 so you have $15
Day 6 you save $6 so you have $21
```

Question 4 (25%)

Write your code in *Q4( )* that displays the triangular pattern below according to the input size and first number of each row. Both input values are **single-digit positive integers**, and the first number is not less than size of the pattern. You need to use **nested loops** in your answer.

6th digit of Student ID	**Pattern …**	**Symbol**
Odd	Inverted	*
Even	Upright	#

Sample display (for odd)

```
Size: 5
First number: 7
7*6*5*4*3*
  7*6*5*4*
    7*6*5*
      7*6*
        7*
```

Sample display (for even)

```
Size: 5
First number: 7
        7#
      7#6#
    7#6#5#
  7#6#5#4#
7#6#5#4#3#
```

More examples

Input	Output **(for odd)**	Output **(for even)**
`Size: 1` `First number: 1`	`1*`	`1#`

`Size: 3`
`First number: 9`
	`9*8*7*`
	`  9*8*`
	`    9*`
		`    9#`
		`  9#8#`
		`9#8#7#`

`Size: 4`
`First number: 4`
	`4*3*2*1*`
	`  4*3*2*`
	`    4*3*`
	`      4*`
		`      4#`
		`    4#3#`
		`  4#3#2#`
		`4#3#2#1#`

Note: There should be **NO blank line** in the output. You should utilize repetition structure and NOT <*iomanip*> library functions for this question.

Submission

You are required to **insert your C++ code into the given template file**, and submit the **final** source file to Moodle before the deadline. Use your student name and ID as the filename: ***StudentID_Name.cpp***. Remove all spaces, hyphens and other non-letter characters in the filename. A correct filename should look like: 12345678A_ChanTaiMan.cpp.

Grading

Your program (i.e. the template file with your answers) will be executed by script with different test cases in **Microsoft Visual Studio** using the **Release** setting. The tester will execute the program and enter the question number in "Program Selection Menu" to test a particular question. The program will be restarted for testing each question individually.

You need to follow **EXACTLY** the above input and output requirements. Any deviation from the requirement is considered as incorrect and **no mark** is given for that test case.

Late submission: 100% deduction. **No late submission is allowed**. Submit your work to Moodle some time ahead of the deadline. Late submissions due to slow internet speed will not be accepted.

Syntax error: 5% - 20% deduction depends on the seriousness of the syntax error. You will get **0 mark** if your program contains too many syntax errors. Check your final source file using Microsoft Visual Studio (not those online compliers) carefully before submission.

Runtime error: No mark for the particular test case that triggers the runtime error (e.g. infinite loop, divide by zero, etc.).

Logic error (bug): No mark for the particular test case that deviates from the requirement. Note that a logic error may lead to **failure in ALL test cases** of a question, e.g. displaying incorrect messages, incorrect spelling and spacing, incorrect number format, or incorrectly decide the odd/even version, etc.

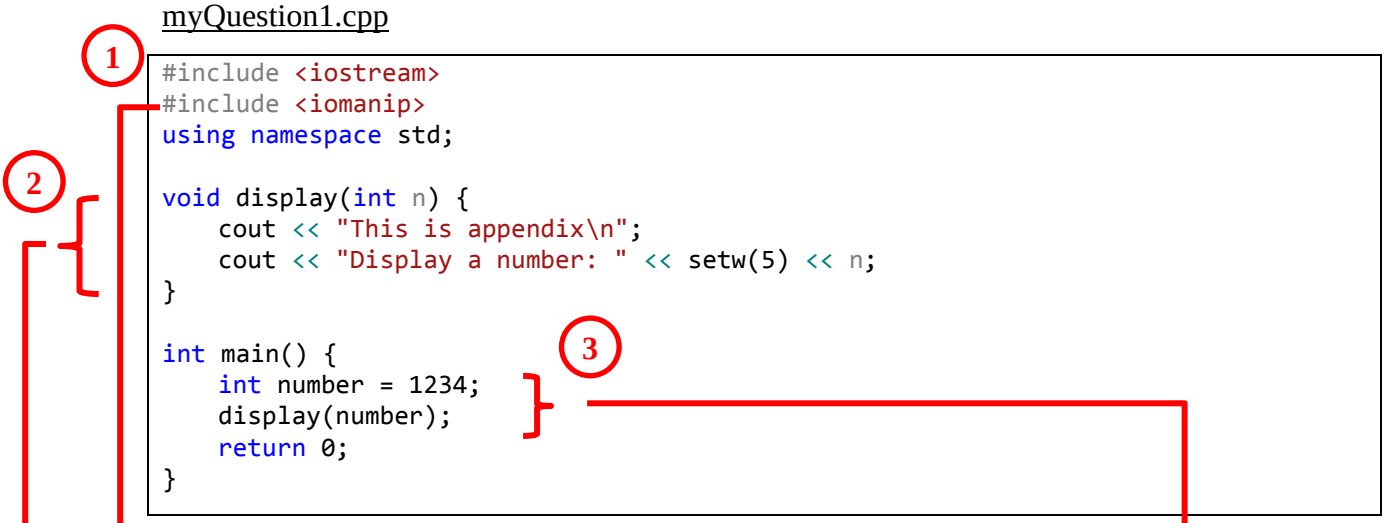
Ensure the originality of your work. Plagiarism in any form is highly prohibited.

- End -

Appendix

If you implement the questions in separated source files, you need to copy the program codes into the template file for assignment submission. Make sure to **test the final source file (i.e. template file with your answers)** in Microsoft Visual Studio before submission.

myQuestion1.cpp

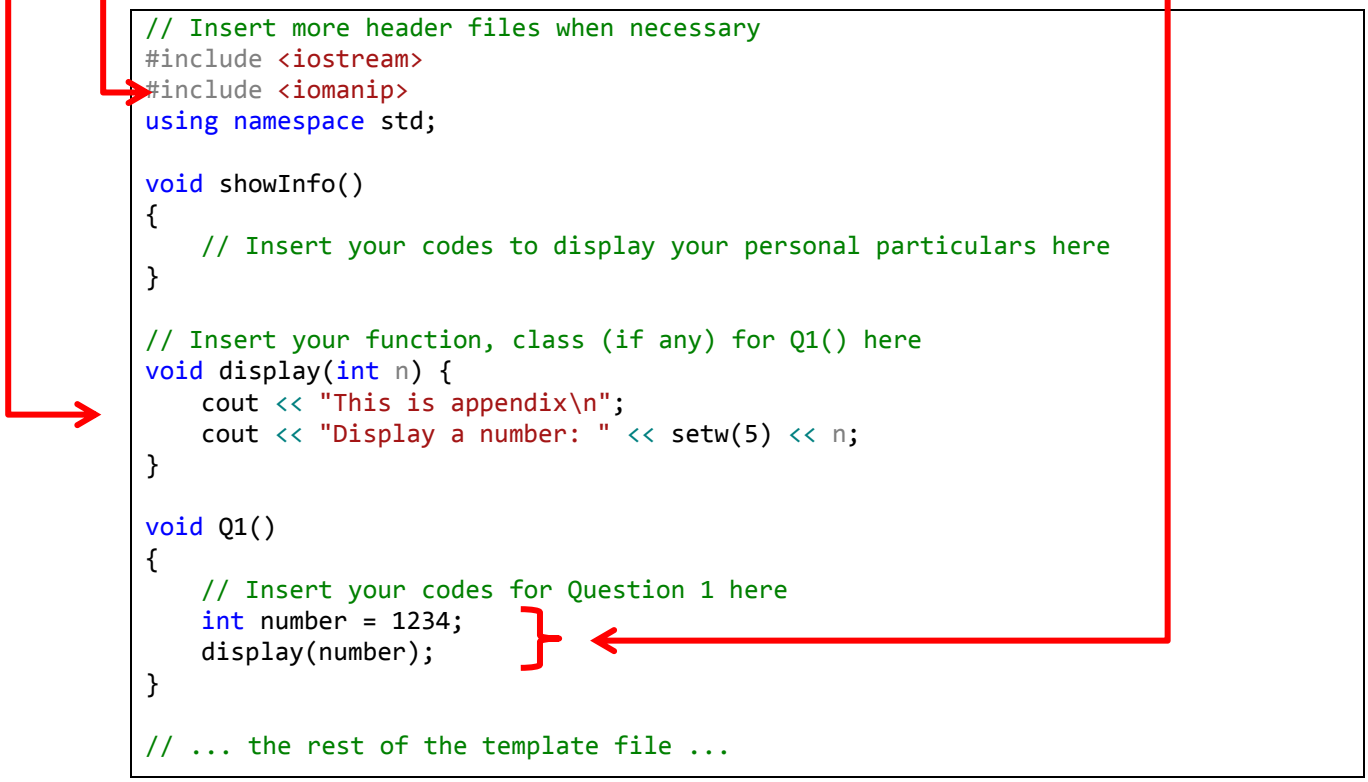


```cpp
#include <iostream>
#include <iomanip>
using namespace std;

void display(int n) {
    cout << "This is appendix\n";
    cout << "Display a number: " << setw(5) << n;
}

int main() {
    int number = 1234;
    display(number);
    return 0;
}
```

Template.cpp



```cpp
// Insert more header files when necessary
#include <iostream>
#include <iomanip>
using namespace std;

void showInfo()
{
    // Insert your codes to display your personal particulars here
}

// Insert your function, class (if any) for Q1() here
void display(int n) {
    cout << "This is appendix\n";
    cout << "Display a number: " << setw(5) << n;
}

void Q1()
{
    // Insert your codes for Question 1 here
    int number = 1234;
    display(number);
}

// ... the rest of the template file ...
```

1. The **header files** included in your program should also be included in the template file.
2. The **user-defined function / class** for a question should be copied **before** the question.
3. The program **main body, except "return 0"**, should be copied to the function body of the corresponding question.