

CCN2042 Computer Programming

Assignment 1 – Individual assignment

Submission deadline: 18:00, 13 Oct 2017

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 *AltTemplate.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

- In the given program template file, you should insert code in the *showInfo* function so that the program displays your personal particulars in the following format when it begins the execution:

```
Name      : XXX YYZ ZZZ
Student ID: 16xxxxxxA
Class     : 101A
```

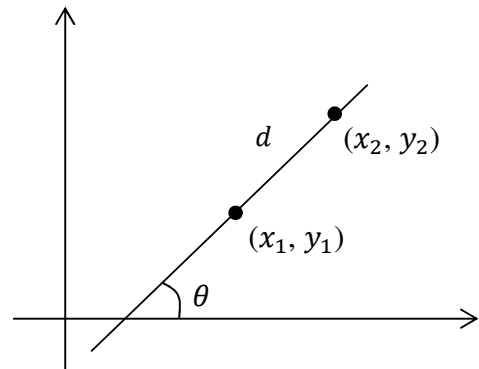
- To answer the following questions, you are allowed to insert codes into the functions *Q1*, *Q2* and *Q3*. You can also include more header files in the program if necessary.
- You are **NOT** allowed to modify any codes in the *main* function. You are **NOT** allowed to add more functions in the program.
- You are **ONLY** allowed to use the C++ programming techniques under the topics up to “Functions (Part 1)” from CCN2042 for the inserted codes in this assignment.
- Whenever a print message is required, you need to follow **EXACTLY** the message format and requirement as stated in the corresponding question.

Question 1: Coordinate Geometry

Consider any two points (x_1, y_1) and (x_2, y_2) on a 2-D coordinate plane, we can calculate the distance between these 2 points and the angle that the line makes with x-axis using the equations below:

1. Distance: $d^2 = (y_2 - y_1)^2 + (x_2 - x_1)^2$

2. Angle: $\tan \theta = \frac{y_2 - y_1}{x_2 - x_1}$



In this question, which equation you should use depends on whether the **6th** digit of your student ID number is *odd* or *even*, according to the table below:

If the 6th digit of your student ID number is ...	You should calculate ...
Odd	Distance
Even	Angle

Complete the function *Q1*, so that when the user enters 1 in the Program Selection Menu, the program performs the calculation with the following requirements:

- 1R1 Use the messages below to prompt for the user's input of the coordinates of two points. The user inputs the x- and y-coordinate of a point in the same line separated by a space:

Input (x1, y1): 3.2 1.5

Input (x2, y2): 17.5 14.2

- 1R2 According to the equation you use in this question, the program calculates either *distance* or *angle*, and displays the result using the following output messages (e.g. input coordinates as shown in 1R1):

The distance is 19.125 units

or

The angle is 0.726 radians

- 1R3 Following 1R2 above, the calculation result should be displayed in 3 decimal places.

- 1R4 After showing the calculation result, question 1 completes and the program returns back to the Program Selection Menu.

- 1R5 Make sure that your program implements the correct equation (based on your student ID number), and the calculation is correct.

- 1R6 You can assume that the users always enter numbers as the input for all the coordinates.

Question 2: Number Extraction

In this question, a positive integer value is considered. By checking every 2 consecutive digits from the value, you are going to find the *largest* or *smallest* 2-digit value according to the table below:

If the <u>7th</u> digit of your student ID number is ...	You should find ...
Odd	Largest 2-digit value
Even	Smallest 2-digit value

Complete the function *Q2*, so that when the user enters 2 in the Program Selection Menu, the program finds the required value with the following requirements:

- 2R1 Use the messages below to prompt for the user's input of any positive integer value with 2 to 9 digits:

Input a positive integer with 2 to 9 digits: 462056843

- 2R2 Following 2R1 above, if the user inputs an invalid value, the following error message should be displayed, and the program should allow user to input again until a correct value is received:

Error: Out-of-range. Input again: 4
Error: Out-of-range. Input again: 1462056843
Error: Out-of-range. Input again: -46205
Error: Out-of-range. Input again: 462056843

- 2R3 According to the value you find in this question, the program displays the 2-digit result using the following output messages (e.g. input value as shown in 2R1):

The largest 2-digit value is 84
or
The smallest 2-digit value is 05

- 2R4 After showing the result, question 2 completes and the program returns back to the Program Selection Menu.

- 2R5 Make sure that your program finds the correct 2-digit value (largest or smallest) based on your student ID number.

- 2R6 Appropriate use of loop is required in your codes.

- 2R7 You can assume that the users always enter integers when being asked for input, and the first digit of the input number is a non-zero value.

Question 3: Cross

In this question, you are going to print blocks of stars according to the size input by the user. The pattern to be generated depends on whether the 8th digit of your student ID number is *odd* or *even*, according to the table below:

If the <u>8th</u> digit of your student ID number is ...	The program prints ...
Odd	A square cross pattern
Even	A x-shaped cross pattern

Complete the function *Q3*, so that when the user enters 3 in the Program Selection Menu, the program prints the required pattern with the following requirements:

- 3R1 Use the messages below to prompt for the user's input of size within the range 1 to 10, inclusively:

```
Input size (1 to 10): 4
```

- 3R2 Following 2R1 above, if the user inputs an invalid size, the following error message should be displayed, and the program should allow user to input again until a correct size is received:

```
Error: Out-of-range. Input again: 13
Error: Out-of-range. Input again: 0
Error: Out-of-range. Input again: -5
Error: Out-of-range. Input again: 4
```

- 3R3 Note the meaning of size in the pattern. Make sure that the number of rows and the number of stars in each row are printed correctly.
- 3R4 After printing the pattern, question 3 completes and the program returns back to the Program Selection Menu.
- 3R5 Make sure that your program prints the correct pattern (square cross or x-shaped cross) based on your student ID number.
- 3R6 Appropriate use of nested loop is required in your codes.
- 3R7 You can assume that the users always enter integers when being asked for input.

Example printouts of the cross pattern

Size	Square cross pattern	X-shaped cross pattern
1	* *** *	* * * * *
2	** ** ***** ***** ** **	** ** ** ** ** ** ** ** ** **
3	*** *** *** ***** ***** ***** *** *** ***	*** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
8	***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** *****	***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** *****

Submission

This is an individual assignment. You are required to submit the **final** C++ source code file, which contains your codes for Questions 1 to 3 inserted into the template file provided. Use your student name and ID as the filename: *StudentName_ID.cpp*. Submit the file to Moodle before the deadline: **18:00, 13 Oct 2017 (Friday) (Week 6)**.

Late submission is subject to 20% deduction in your final marks for each day late (including public holiday and Sundays). No late submission is allowed **4 days** after the due date.

Grading Criteria

	Marks allocation (out of 100 marks)	Grading aspects	Percentage
ShowInfo	3	Correctness	100%
Question 1	30	Correctness	90%
		Program design	10%
Question 2	30	Correctness	80%
		Error input checking	10%
		Program design	10%
Question 3	30	Correctness	80%
		Error input checking	10%
		Program design	10%
Program standard, comments, line spacing, program clarity, indentation	7	n/a	n/a

Marks deduction

Marks will be deducted if the program fails to be compiled. The deduction is from 5 to 20 marks, depending on how serious the compilation error is. Note that if the program contains unacceptably too many serious compilation errors, your program will score 0 marks. Compilation errors also lead to failure in the program correctness if the function cannot be tested.

Failure to decide the program requirements correctly based on your student ID would lead to marks deduction under program correctness.

You should make sure your program can be compiled and run properly in our standard program testing environment, i.e., by Microsoft Visual Studio 2010 Professional.

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

- End -