

Programming Evaluation

Module 1 – High-level Programming 1

Copyright Notice

Copyright © 2017 DigiPen (USA) Corp. and its owners. All rights reserved.

No parts of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language without the express written permission of DigiPen (USA) Corp., 9931 Willows Road NE, Redmond, WA 98052

Trademarks

DigiPen® is a registered trademark of DigiPen (USA) Corp.

All other product names mentioned in this booklet are trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Purpose

The purpose of this assignment is for you to learn how to use expressions, conditionals, and iterations in order to solve a specific problem and generate the required output.

Information

Write a program that allows students to draw the following shapes in the console:

- Filled rectangle
- Non filled rectangle
- Filled triangle
- Non filled triangle

The program will start by giving the user the option to pick between five choices:

- 1 - A Filled Rectangle
- 2 - A Non Filled Rectangle
- 3 - A Filled Triangle
- 4 - A Non Filled Triangle
- Any Other Number - Exit

Based on the user's choice, the program will ask the user to input one or more values (rectangle width and height, triangle height) that the program will use to draw the shape correctly.

Once the shape is drawn, the user is given the options again. This process will only end when the user inputs a number that is not one of the choices.

Sample Runs (user's input in red):

```
what would you like to draw?
1 - A Filled Rectangle
2 - A Non Filled Rectangle
3 - A Filled Triangle
4 - A Non Filled Triangle
Any Other Number - Exit

Your choice is: 1
Enter the rectangle's width and height values: 10 3
*****
*****
*****

what would you like to draw?
1 - A Filled Rectangle
2 - A Non Filled Rectangle
3 - A Filled Triangle
4 - A Non Filled Triangle
Any Other Number - Exit

Your choice is: 0
Thank you for using our application!
```

```

What would you like to draw?
1 - A Filled Rectangle
2 - A Non Filled Rectangle
3 - A Filled Triangle
4 - A Non Filled Triangle
Any Other Number - Exit

Your choice is: 1
Enter the rectangle's width and height values: 0 5
The rectangle width has to be at least 1

What would you like to draw?
1 - A Filled Rectangle
2 - A Non Filled Rectangle
3 - A Filled Triangle
4 - A Non Filled Triangle
Any Other Number - Exit

Your choice is: 0
Thank you for using our application!

```

```

What would you like to draw?
1 - A Filled Rectangle
2 - A Non Filled Rectangle
3 - A Filled Triangle
4 - A Non Filled Triangle
Any Other Number - Exit

Your choice is: 4
Enter the triangle's height value: 5
  *
 * *
*   *
*   *
*****

What would you like to draw?
1 - A Filled Rectangle
2 - A Non Filled Rectangle
3 - A Filled Triangle
4 - A Non Filled Triangle
Any Other Number - Exit

Your choice is: 0
Thank you for using our application!

```

Note: The user's input will not show in the output files!

You'll need to pay close attention to the alignment and formatting since your output has to exactly match the provided sample output files.

Note: An [executable](#) will be provided in order for you to test values and check out what the appropriate output should be.

You are to use these input values to test your program:

Type		Input Value	Output file
Filled Rectangle Tests Only	Test 1	1 10 3 0	output_sample_filled_rectangle_1(1-10-3-0).txt
	Test 2	1 3 10 0	output_sample_filled_rectangle_2(1-3-10-0).txt
	Test 3	1 1 1 0	output_sample_filled_rectangle_3(1-1-1-0).txt
	Test 4	1 1 5 0	output_sample_filled_rectangle_4(1-1-5-0).txt
	Test 5	1 0 5 0	output_sample_filled_rectangle_5(1-0-5-0).txt
	Test 6	1 5 -3 0	output_sample_filled_rectangle_6(1-5--3-0).txt
Non Filled Rectangle Tests Only	Test 1	2 10 3 0	output_sample_non_filled_rectangle_1(2-10-3-0).txt
	Test 2	2 3 10 0	output_sample_non_filled_rectangle_2(2-3-10-0).txt
	Test 3	2 1 1 0	output_sample_non_filled_rectangle_3(2-1-1-0).txt
	Test 4	2 1 5 0	output_sample_non_filled_rectangle_4(2-1-5-0).txt
	Test 5	2 0 5 0	output_sample_non_filled_rectangle_5(2-0-5-0).txt

	Test 6	2 5 -3 0	output sample non filled rectangle 6(2-5--3-0).txt
Filled Triangle Tests Only	Test 1	3 5 0	output sample filled triangle 1(3-5-0).txt
	Test 2	3 1 0	output sample filled triangle 2(3-1-0).txt
	Test 3	3 -3 0	output sample filled triangle 3(3--3-0).txt
Non Filled Triangle Tests Only	Test 1	4 5 0	output sample non filled triangle 1(4-5-0).txt
	Test 2	4 1 0	output sample non filled triangle 2(4-1-0).txt
	Test 3	4 -3 0	output sample non filled triangle 3(4--3-0).txt
Other Tests	Test 1	0	output sample other 1.txt
	Test 2	1 10 5 3 7 4 2 3 -1 0	output sample other 2.txt

When you're ready to compare your output to the expected output, you'll need to redirect your output to a file (e.g. `myoutput.txt`) and then, using a diffing tool (such as WinMerge) compare your output with the expected output.

Testing your code

Sample output files are provided in order for you to check if your implementation is correct.

In order to run the tests follow these steps:

- Take a look at the various files in the Sample Outputs folder. In cases where there are multiple possible tests or inputs for those tests, the text files will be labeled accordingly.
- Adjust your main.cpp file so it's running one of the tests in the Sample Outputs folder with matching inputs if necessary. Redirect the output of your program to a text file. You can do this by running the following command:
 - myExecutable > myOutput.txt
- Choose a diff tool such as “diff” (available via MinGW, Cygwin, or on the CS50 IDE), [WinMerge](#), or a website such as <https://text-compare.com/>
- Compare the corresponding file from the Sample Output folder to the output from your program.

What to submit

You must submit the CPP file (**main.cpp**) in a single .zip file (go to the class Canvas page and you will find the submission link).

Do not submit any other files than the ones listed.