

CSE108 – Computer Programming Lab.

Lab 4

Loop, Functions, File Operations

Due 31/03/2023 at 10am.

Hand in: A student with number 20180000001 should hand in a zip file named 20180000001.zip for this lab.

In this lab assignment, you are required to create a simple program that reads and prints basic shapes. The program should read a sequence of shapes and their corresponding size attributes from a text file, and then draw them on the console using '*' characters. The shapes can be one of two types: square, or triangle. You must write a drawing function for each shape type, as well as a function that reads user input for desired shapes and their attributes and writes them to a text file.

In the main function, the program should read the sequence of shapes from the text file line by line and call the drawing function of the shape at current line until all shapes have been drawn. **It is important to note that you are not allowed to program any part of this assignment in the main function if it has been specified that it must be done as a separate function. Using arrays are forbidden in this assignment.**

At the start of the program, a menu will be displayed to the user. Once the user has generated a shape text file, the second option in the menu becomes active, otherwise it has to be irresponsive when it is selected. After all the shapes have been drawn, the menu reappears.

The proper version of the main function is worth **20 points**. See below for details on the required functions.

Menu:

```
Welcome to Shape Reader! Please make your choice to continue:
1-) Generate a shape file
2-) Read and draw a shape file
3-) Terminate the program
```

Part 1. (20 pts) Write sequence of the shapes to file: This function prompts the user for input of a shape type (square: 's', triangle: 't') and an integer size attribute (between 3 - 10) for the shape. It then writes the shape into a text file, one at a time, until the user enters 'e' as the shape type to indicate the end of the writing process. Each line in the text file contains the data for a single shape, with the data separated by commas. An example of the contents of a shape file is provided below:

t,5

s,4

s,6

t,7

Part 2. (30 pts) Draw Square: This function takes the size of a square as input and draws it on the terminal. The size parameter specifies the edge length of the square. The function uses the '*' character to draw the borders of the square and the space character to draw empty spaces inside the square. A space should be printed after each star to prettify the drawing. For example, a square with a size of 4 should be drawn as follows:



Please note that the borders are included to the size for all the types of shape.

Part 3. (30 pts) Draw Triangle: This function takes the size of a equilateral triangle as input and draws it on the terminal. The size parameter specifies the edge length of the triangle. The function uses the '*' character to draw the borders of the equilateral triangle and the space character to draw empty spaces inside the triangle. A space should be printed after each star to prettify the drawing. For example, a triangle with a size of 7 should be drawn as follows:

Example triangle with size 7:



The bottom edge of the triangle should have more stars than the sides, so please add extra stars as needed. The side edges of the triangle must have the same number of stars as the size of the triangle.

General Rules:

1. You will have two hours to provide a solution to the given problem set. You are not permitted to ask any questions. If there is a significant error in the assigned tasks, it will be addressed later.
2. You will be able to hand in your solutions via Teams in the next two hours. The submission will be closed exactly at 10am.
3. There will be an interview session immediately after the submission deadline. Starting at 10am, you will be randomly invited to attend a meeting by a TA to demonstrate your solution and answer any questions asked by the TA.
4. You must be available until 1pm to respond to the demo invitation whenever you receive it. You will have 3 minutes after you are called via Teams. If you do not answer/appear in 3 minutes, you will miss your interview.
5. If you miss your interview or are unable to give satisfactory answers to the questions, you will receive a zero for that lab even if you have submitted your solution.

6. If you have not submitted a solution in time, you will not be invited for the interview and receive zero for that lab.
7. Due to time constraints, some students may not be invited to an interview. In that case, their solutions will be graded offline.
8. Unless you aren't declared for a specific prototype, you may use arbitrary but proper function and variable names that evoke its functionality.
9. The solution must be developed on given version of OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won't be tolerated.
10. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it is working partially.
11. Zip your solution file before uploading it to MS Teams. The zip file must contain the C file with your solution and screenshots of the valid outputs of the program.