CS 135

Programming Assignment 0B (PA0B-09/09)

You must turn your programming assignment in by 6:00 pm Monday, 09 September. Like the first one, this is an easy assignment but you should try to get it finished as soon as possible so you can get help if you need it.

Objectives:

1) You will use an Integrated Development Environment (IDE) to create and run given programs
2) You will correctly modify and redevelop program code that demonstrates sequential or linear progression
3) you will read and correctly analyze or describe program code that demonstrates sequential or linear progression
4) You will modify and develop program code in a block-structured format that effectively separates differing program actions such as input/output (I/O) and different processing tasks from each other in an appropriate modular form
5) You will use simple **`void`** functions to solve problems and/or meet given specifications in a computer program

Tasks:

Updating a Previous Program

1) This assignment requires you to take the previous code you developed for the previous assignment and slightly modify it.

2) Create a new folder for your Week 2 activities.

3) Download the *base_1B.cpp* program, then change the name of the file to *par_1B.cpp*, and write the code so that the program displays a hollow parallelogram. For this exercise you will be using the *printAsterisk*, *printSpace*, and *printEndLine* functions to display the individual items on the screen. These are the only functions provided to you in this program and you may not modify them or add any other supporting code. All program code you write must be within the *main* function. Also, you must have one comment line in the code (prior to the code statements) for each full line displayed on the screen, as was provided in the first programming assignment.

4) The parallelogram must look like the following:

```
**************
**************
****          **
***          ***
**          ****
**************
**************
```

5) Now, using the same *base_1B.cpp* program, save the file as *tri_1B.cpp*, and write the code that displays a hollow triangle that looks like this one.

```
***********
***********
*****  *****
****    ****
***      ***
**        **
***********
***********
```

6) Finally, using the same *base_1B.cpp* program, save the file as *di_1B.cpp,* and write the code that displays a hollow diamond that looks like this one.

```
***********
***********
*****  *****
****    ****
***      ***
**        **
*          *
**        **
***      ***
****    ****
*****  *****
***********
***********
```

7) Now download the *base_2B.cpp* program, and create a parallelogram, a triangle, and a diamond that look like the ones provided previously in this document. You will need to create individual files for each (*par_2B.cpp*, *tri_2B.cpp*, and *di_2B.cpp*). For this exercise you will be using the *printAsterisks*, *printSpaces*, and *printEndLines* functions to display the individual items on the screen. As you may recall from the previous assignment, these functions require parameters, or values in them that specify how many of the asterisk, space, or endline to print. You may not use numbers in these parameter lists; you must use constants

and/or variables. In general, you will not use numerical values in parameter lists very often; they do not support good code readability. These specified functions are the only functions provided to you in this program and you may not modify them or add any other supporting code. All program code you write must be within the *main* function. Again, remember that you must have one comment line in the code (prior to the code statements) for each full line displayed on the screen.

8) Finally, download the *base_3B.cpp* program ***and*** the *formatted_console_io_v19.h* header file as you did in the first laboratory. The header file must be in the same location as the program file but you must save it as header file instead of a source code (i.e., .cpp) file. Once these are downloaded, create a parallelogram, a triangle, and a diamond that look like the ones provided previously in this document. You will need to create individual files for each (*par_3B.cpp, tri_3B.cpp,* and *di_3B.cpp*). Display the parallelogram left justified on the center of the screen; display the triangle right justified on the center of the screen; and display the diamond center justified on the center of the screen. For this exercise you will be using only the *printStringAt* function as you did previously for the third part of your first assignment. As you may recall from the previous assignment, this function also requires parameters; you may place the string literal part (e.g., "*****") in the function parameter list, and you will need to place the "LEFT", "RIGHT", or "CENTER" string literals as needed in the parameter list, but the x and y locations of the strings must use either constants and/or variables. To repeat, with only a few exceptions, it is not a good idea to place numbers in a parameter list. This specified function is the only function needed in this program and you may not modify it or add any other supporting code. All program code you write must be within the *main* function. Again, remember that you must have one comment line in the code (prior to the code statements) for each full line displayed on the screen.

9) Create a screenshot for all nine programs. Each group of three will look alike but you need to be clear in your annotation which screenshot is displayed from which program. The title for this document must be "ScreenShots".

Turning in your laboratory assignment:

You should assume that this format will be used for all your laboratory assignments throughout the term unless otherwise specified. It is suggested that you use the following as a checklist when you are preparing your assignment.

Note that correctly following these instructions is evaluated as part of your laboratory grade; not following these instructions carefully may result in a loss of credit. In most cases, you must turn in your programs well before the 6:00 pm deadline. ***Last-minute assignment upload failures may still experience the 50% credit reduction even if the problem is due to technical issues***.

Use the WebCampus assignment to upload the following program materials.

1. Collect the following files in one folder on your computer ("Week 2"):

    1. The Word file named “ScreenShots” containing the following:
        1. There should be at least nine (9) screenshots for the program as specified above
        2. Remember to clearly annotate every displayed result

    2. The executable files:
        1. par_1B.exe
        2. tri_1B.exe
        3. di_1B.exe
        4. par_2B.exe
        5. tri_2B.exe
        6. di_2B.exe
        7. par_3B.exe
        8. tri_3B.exe
        9. di_3B.exe

    3. The source code files:
        1. par_1B.cpp
        2. tri_1B.cpp
        3. di_1B.cpp
        4. par_2B.cpp
        5. tri_2B.cpp
        6. di_2B.cpp
        7. par_3B.cpp
        8. tri_3B.cpp
        9. di_3B.cpp

2. Select all of these files, right click on them, and select “Send To”, then select “Compressed (zipped) Folder”. Note that you must use the “zip” file. If you use any other compression form such as .rar, your assignment will not be graded.

3. Once the folder is created, it will be placed in the same folder in which you are working. Change the name of the zipped folder to your own name in the form "LastnameFirstname_PAX" (where X is the Programming Assignment number - the first Programming Assignment is PA0A) as follows: "LeveringtonMichael_PA0A" (no quotes). After you have renamed the zipped folder, double click on it to verify that it has all the files it is supposed to have.

4. Follow the instructions in the previous assignment for uploading, or you can use the reference document, ***How to Upload Your Laboratory Assignments***, found in the ***General Course Information*** folder under the ***Course Information*** topic.