CS 135

Programming Assignment 0 (PA0-09/03)

You must turn your programming assignment in by 6:00 pm next Monday, 03 September. This is an easy assignment and you will have plenty of time to work on it in your first laboratory time but you need to try to finish it as soon as possible; waiting until the due date usually leads to problems.

Note that next Monday is a holiday and classes and office hours will not be held. However, to make the schedule work with next Tuesday's laboratory, the due date must be fixed on Monday. As stated above, this is an easy assignment and you should have no trouble finishing it early.

Note that if the programming assignment is turned in late, but prior to 12:00 m (midnight) the day it is due, the laboratory credit will be reduced by 50% of the earned score. Any laboratories turned in more than 6 hours late will not earn any credit. This is the standard policy for the course.

Objectives:

1) You will use an Integrated Development Environment (IDE) to create and run given programs
2) You will create an organized file system for your programming projects
3) You will correctly develop program code that demonstrates sequential or linear progression
4) you will read and correctly analyze or describe program code that demonstrates sequential or linear progression
5) 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
6) You will use simple `void` functions to solve problems and/or meet given specifications in a computer program

Tasks:

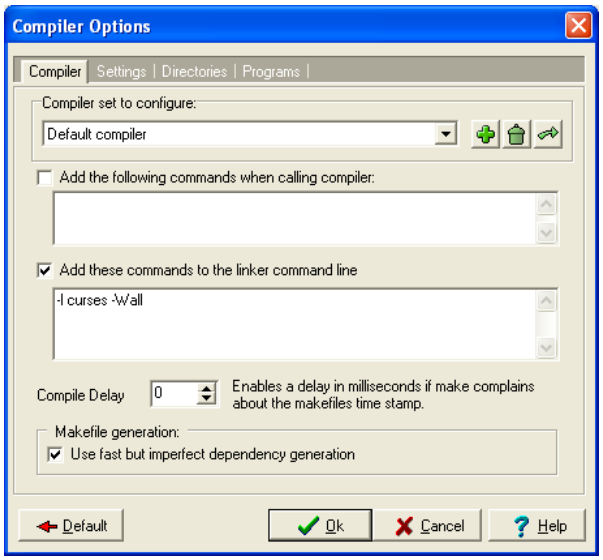
Part I: Configuring the Programming Environment

1) You must acquire a login for the Engineering Computing Center (ECC) before you can work on the laboratory computers. Do this as soon as you can, and try to have it completed by laboratory time, if possible. Your ECC login is NOT the same as your NetID login; you must get this login from the ECC staff and you may only use it in the ECC.

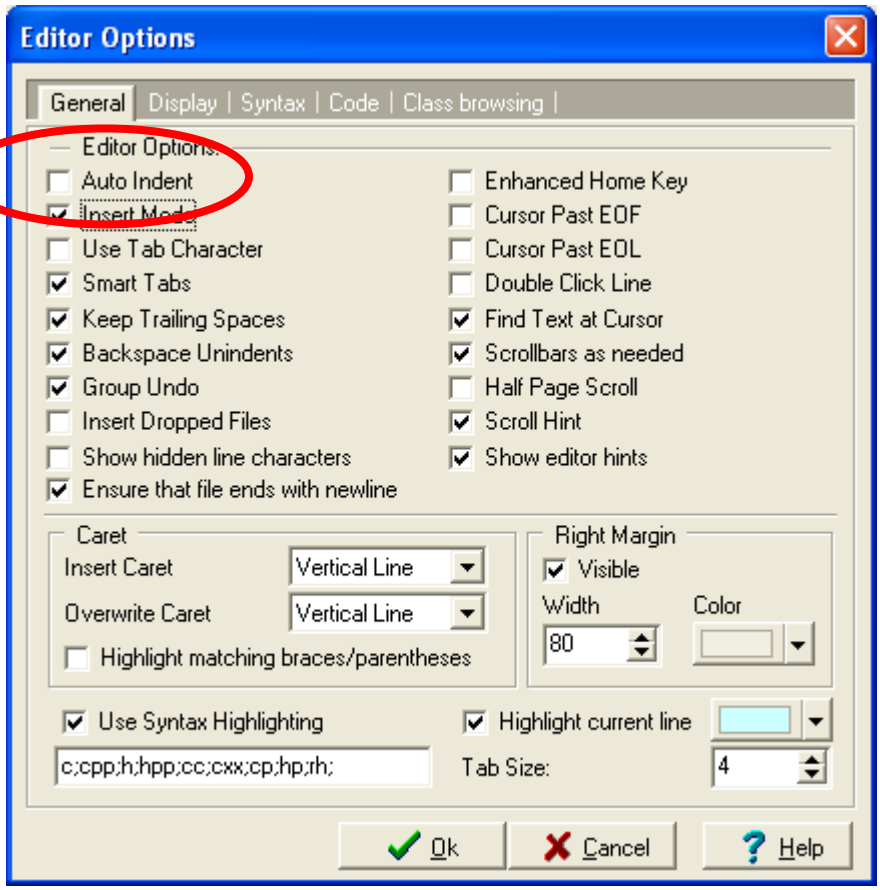
2) Your very first step should be to save this PDF file to your desktop so you can refer to it as you are following the next instructions. You will need to move around the WebCampus environment, so it will not always be possible to keep this file open if you don't save it and open it up separately.

3) Next, if this is your first time with C++ programming in the ECC, you must configure the Dev C++ Integrated Development Environment (IDE). Follow these instructions **carefully** so that you will not have extra baggage to worry about when you are using this IDE during the semester. Double-click on the blue and white "Dev C++" icon on your desktop. Then:
   a) In the first "Dev C++ First Time Configuration" window, select "Next" for the English language which is already highlighted.
   b) Then for the next window that retrieves information from header files, select "No" and select "Next".
   c) The last window tells you that you are finished, so select "OK" and you will be ready to go to work.

4) Next, go to the "Tools" menu and select "Compiler Options". Then:
   a) Check the box that says, "Add these commands to the linker command line"
   b) Type the text "`-l curses -Wall`" (dash, lower case letter: 'L', space, lower case text: "curses", space, dash, upper case letter: 'W', then lower case text: "all") into the text area, as shown below.



5) Finally, go back to the “Tools” menu and select “Editor Options”. Uncheck the “Auto Indent” check box as shown below. This will save you hours of fixing curly brace locations as you are programming.



6) The entire process of downloading and configuring Dev-C++ with Curses is located in the “General Course Information” folder in WebCampus. You do not need to do this in the ECC, but you may use this as needed to download and set up this software on your own computer.

7) If you have any problems using Dev C++, continue in this document to the section on “Reasons you may have problems”.

Part II: Setting up Your File System

1) First, you must find your storage drive. There are several ways to do this, but if you are unfamiliar with the process, double click on your "My Computer" icon on the desktop. You will see three or four drives displayed, but one of them will have "**<yourLogin> (\\carl\Users) (H:)**" (or something equivalent) where **<yourLogin>** is the login name you were given by the ECC staff. Double-click on this icon.

2) The following are suggested ways to manage your files. One way that usually does not work is to save your files on your desktop. Your programming environment is unlikely to work if you use the desktop for your files.

3) Find the “Documents” and “My Documents” folders.

4) Outside of the ECC, using Windows XP. Select (i.e., click your mouse on) the “My Documents” folder, then select the "File" menu at the top left of the screen, then select "Folder".

5) In the ECC, Windows 7. Select (i.e., click your mouse on) the “My Documents” folder, then select "New Folder" near the top of the screen.

6) With either system, a folder icon will be displayed with the title "New Folder" under it. Change "New Folder" to "CS135" and then press enter. Then double-click on the CS135 folder, create another new folder (as you did before), and name this folder "Week 1".

7) While you are working in the ECC, all your activities should be stored in this folder. However, when you are finished in the ECC, you should store your programs to a cloud service such as Google Docs, DropBox, etc. In addition, you should have at least one jump drive (memory stick) with exactly the same folder names so you can back up everything you do to the jump drive, which you should always do. Now either minimize or close the file folder window. You will ONLY have access to your ECC files during normally scheduled ECC operating hours.

8) Note that if you are unfamiliar with file and folder organization, there is a brief but pretty good tutorial on this at http://cter.ed.uiuc.edu/tutorials/filemanagmt/.

Part III: Getting Started with the Programming

1) Dev C++ will be the primary tool that you will be using during the semester, and it should still be open and running on your computer. An IDE is a software tool that lets you 1) edit the program text you are working on, then 2) translate the text into binary code that the computer can use, and then 3) run and test the program that has been created. If you have typed everything according to the rules of C++ programming (i.e., the syntax is correct), your program will compile correctly. If you have not, it should show you the mistakes you have made at the bottom of the screen.

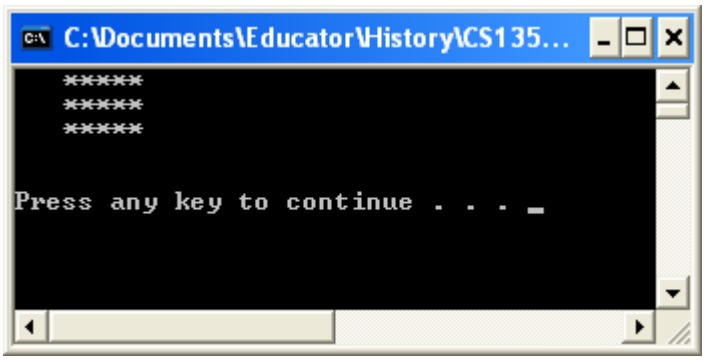
2) Next, if you have not already done so, log in to your WebCampus, go to the "Week 1" folder, then go to "Laboratory - Week 1", and open the file named *Rectangle_1*. It is important to note that this file is actually *Rectangle_1.cpp*, but WebCampus does not show file extensions (i.e., the letters after the period that follow the file name). Select ALL of the text in the file - you may use Ctrl-A (i.e., press the "Ctrl" key and the letter 'A' at the same time) or right-click on the text and select "Select All". Then copy the text - you may use Ctrl-C or you may right-click and select "Copy". Now minimize your WebCampus browser; don't close it yet as you will need it later.

3) Now click on the empty white rectangular icon that is third from the left near the top in Dev C++. This opens an empty text file within which you may work. Finally, paste the data you copied from WebCampus into the text file - you may press Ctrl-V or you may right click and select "Paste". The first thing for you to do is to save this file as "Rectangle_1". Make sure you do not put any spaces into the file name. The file will automatically be saved as a C++ file with the ".cpp" extension, so you don't have to add that.

4) You have now downloaded a program from WebCampus and loaded it into your IDE so you can work on it. This program is fully correct and has quite a few comments in it. Comments are parts of the text file that help you understand the program but are not translated into code by the IDE. After today's laboratory, you should spend some time reading through all of the text in this file. Several parts of the program are explained for you, including the comments themselves. Most of what you see may be new to you at first, but we will be talking about all these components over the next few days.

5) Now that your program is loaded into the IDE, you can "Build" or "Make" the program which means to translate it into a running program. The Dev-C++ calls the process "Compiling", but this is a bit of a misnomer; it is only partially correct. There are three icons near the upper left of the screen, but below the empty text icon that you selected earlier. The first on the far left "Compiles" (i.e., it actually "Builds") the program, which means it translates it into code that the computer can understand and run. The second icon "Runs" the program which should be obvious - it will allow your program to implement its actions. Finally, for your reference, the third icon combines the first two - it Builds and Runs the program with one click. Most of the time you should be using the first “Compile” button.

6) Now that you have a fully functional program loaded into your IDE, build and run the program. You should see a result like the following in a separate command-line window. Keep this window open for the next few steps.

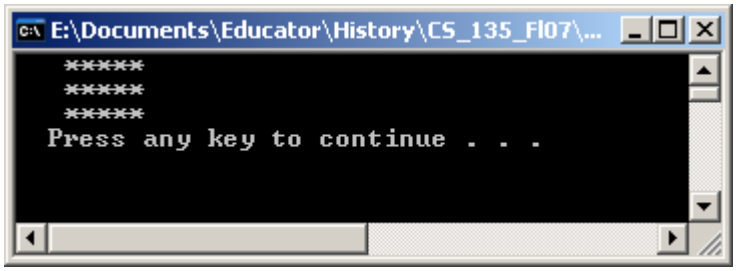


7) In this course you will regularly need to display the results of your programs in the assignments that you turn in. Before you can do this, you must open MS Word on your computer. Once Word is open, implement the following procedure. If you accidentally close the command-line window while you are doing this, just go back to Dev-C++ and run the program again.

8) Go back to the command-line display and use your mouse to shrink the window down so that it is just large enough to show your output; doing this will conserve your toner or ink if you need to print out your results. Now with the command-line window selected (i.e., the title bar of the window is blue, not gray), press the Alt & Print Screen buttons at the same time. The Print Screen button will probably be minimized to something like "PrtScn" or the equivalent, but it is usually located in the upper center to right part of the keyboard. Once you have done this, go back to Word and paste the image into your document. A copy of the command-line window with your results will be displayed.

9) In addition to copying a screen shot as you did above, you ***must always*** annotate <u>each</u> of your output displays so that it is clear what output you are presenting. Here is an example of what you might have by this point:

Example of annotated output presentations:
==========
Here is the rectangle output created by Rectangle_1.cpp: <= Annotation



==========

10) When you are asked for screen shots, you must provide both the picture of your program's display and some information about how it was run, what data was used, what conditions existed, etc (i.e., the "annotation").

11) When you are finished with taking the screen shot, close the program output (command-line) window. You must close this window, in order to compile or run another program. If you do not close it, your compile/run buttons will be grayed out.

12) Once you have run the above program and observed the output, look at the code in the program. You will see that each space and each asterisk is printed as a result of specific commands, which are **`printAsterisk`** and **`printSpace`**. There is also a **`printEndLine`** function so that you can end the line you are on and start over on another.

13) You will also notice that the code is broken into sections or modules, with each module focused on one primary action. In this case each primary action involves the display of one line of output. This organization is called "block structure" or "modularity" and it is critical to effective programming.

14) Save your program to the name *Triangle_1.cpp*. Then, maintaining the code modularity, change the program so that it creates a small triangle, as shown:

```
^^^^*
^^^***
^^*****
^*******
```

The spaces preceding the triangle are shown as carats (^) for your reference. Where you see a carat, you will make a space - DO NOT PRINT THE CARATS. Once you have created the triangle, save (and annotate) the display results in your Word document.

15) Note also that for all the programs in this assignment, you may not change the functions or add your own program code. You must use the **`printAsterisk`**, **`printSpace`**, or whatever functions you are provided for each part of the assignment - some slightly different ones will be provided later in this text. Modifications of the given functions or code are likely to cause a reduction of credit.

16) Now save your program to the file called *Diamond_1.cpp*. Then, again ***maintaining the code modularity*** (i.e., all the code for one displayed output line should be in one block under a descriptive comment), change the program so that it creates a diamond shape, as shown:

```
^^^^*
^^^***
^^*****
^*******
^^*****
^^^***
^^^^*
```

Note that the carats still represent spaces in your program - again, DO NOT PRINT THE CARATS and do not change the function code. Copy your output to your Word document and annotate it.

17) Up to this point, you have been using functions without parameters. This is a little arduous since you have to write a function statement for every space or asterisk. To make life a little easier, there are two more functions you can use called **`printAsterisks`** and **`printSpaces`**. There are two differences between these and the functions you have used thus far. First, they are spelled differently (i.e., they are plural). Secondly, they require a number - called a *value* or *copy parameter* - between the parentheses that were empty in the previous functions.

18) What do you suppose the following code does? Think about the answer for a minute and then go on to the next step.

```
// First Line
printSpaces( 3 );
printAsterisks( 5 );
printEndLine();

// Second Line
printSpaces( 3 );
printAsterisks( 5 );
printEndLine();

// Third Line
printSpaces( 3 );
printAsterisks( 5 );
printEndLine();
```

19) Save and close your present file (menu: "File-Save" and "File-Close"), and open a new empty file using the blank white icon near the top left. Now open the file *Rectangle_2.cpp* in WebCampus, copy the text to your empty file, save it as *Rectangle_2.cpp*, and compile and run it to verify your hypothesis. Copy the resulting output to your Word document and annotate it.

20) Important Note: Notice the underscore in the file name **`Rectangle_2.cpp`** above; this is important. **Never** create a C++ file with spaces in the file name. If you name your file with something like **`Rectangle 2.cpp`** (i.e., with a space), it might actually work sometimes, but most of the time it will cause really weird conditions while you are trying to program, and many times it won't work at all. Build and run the program, and copy and save the results to your Word document, and then annotate it.

21) Once you have built and run the rectangle program, save your program as *Triangle_2.cpp* and use the two new functions to create the triangle picture -- as shown in step 14 -- in place of the rectangle picture. Build and run the program, and copy and save the results to your Word document, and then annotate it.

22) Once the triangle program is working, save your program as *Diamond_2.cpp* and then, using these two new functions, create the diamond picture as shown in step 16, copy it to your Word document, and annotate it.

23) Make sure you save your **`Diamond_2.cpp`** file, and then close this file (menu: "File-Close"). Now open a new empty file using the blank white icon below the menu line. Go back to your WebCampus "Week 1/Laboratory Week 1" folder, and open the **`Rectangle_3.cpp`** file. Copy the entire contents of this file into your empty file and save it as **`Rectangle_3.cpp`**.

24) Next, open another empty file, and copy the contents of **`formatted_console_io_v17.h`** into this file. Select the "File", "Save As…" menu items, and in the "Save File" window, type "formatted_console_io_v17" (no quotes) in the "File name:" window, and then in the "Save as type:" window, select the "Header files (*.h,*.hpp,*.rh,*.hh)" drop down. Then select the "Save" button to save this file as a header file. If you do not do this correctly, your programming process will not work. If this occurs, you must close the "formatted_console_io_v17.h" file in Dev C++, then delete this file in your folder, then download the text from WebCampus and repeat the process.

25) Compile and run the Rectangle_3 program (not the **`formatted_console_io_v17.h`** file; it won't compile). In this case, you will see that three rectangles are displayed. However, the rectangles are located in different places than the previous ones. Take a look at the code and figure out how this is done. There is a generalized function called **`printStringAt`** that takes four parameters, as follows (and in this exact order): 1) the x position on the screen (which is 80 characters wide), 2) the y position on the screen (which is 25 characters tall), 3) the actual series of characters (called a string) that are to be printed, and 4) the justification of the string related to the x position (e.g., left, right, or centered). There are three rectangles located on the screen so that you can see how each of these can be used.

26) Once you have looked over how the **`Rectangle_3.cpp`** code works, create a **`Triangle_3.cpp`** program and a **`Diamond_3.cpp`** program. Unlike the **`Rectangle_3.cpp`** program, ***you only need to create one triangle or diamond***, but you may use any justification format you wish to use. Make a copy of the output display for all three of these programs, annotate them, and place them in your Word document.

Reasons you may have problems:

1. You didn't set up the compiler switches correctly (**`-l curses -Wall`**)
    a. must be a "dash el" (dash and then lower case 'L'), not a one (1)
    b. must have spaces where specified

2. You used a space in the source code file name (**`my file.cpp`**)
    a. resave your source code file “as” a file name without any spaces

3. You tried to save source code to the desktop, instead of the server drive
    a. resave your source code file in the directory under “Documents” and “My Documents”

4. You saved the **`formatted_console_io_v17.`**h file as a .cpp file
    a. close the file window and delete the file completely
    b. open new text window
    c. save file using correct "header - .h" "Save As:" format

5. You tried to recompile and/or run without closing previous command-line window
    a. find and close the command-line window that has the running program

6. You didn't correctly download curses with package manager; this should not be a problem with ECC computers but might be a problem with your own
    a. Select the “Tools” menu items and then the "Package Manager" sub menu item
    b. Look for the “pdcurses” package in the list of icons displayed
    c. If the curses package is in the list, select it, and then check for information in the text boxes on the left hand side of the screen
    d. if the pdcurses file is not there or doesn’t show any information in the text boxes, follow the instructions in the “Dev C++ Installation Process” file that can be found in the “General Course Information” folder in WebCampus

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.

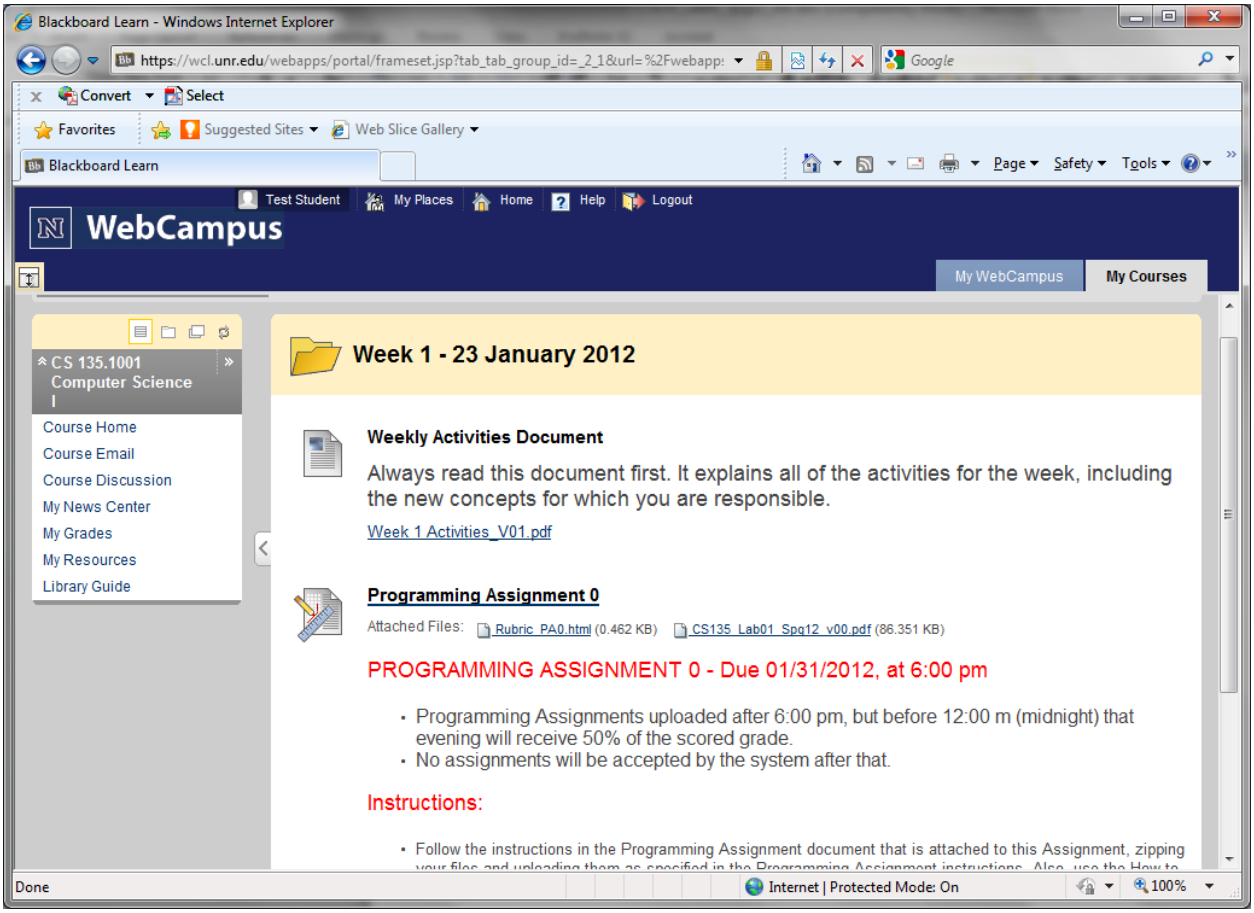
Note that correctly following these instructions is evaluated as part of your laboratory grade; not following these instructions carefully may result in a loss of credit. In most cases, you must turn in your programs well before the 6:00 p.m. deadline. ***Last-minute assignment upload failures may still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

Use the “Programming Assignment 0” to upload the following program materials.

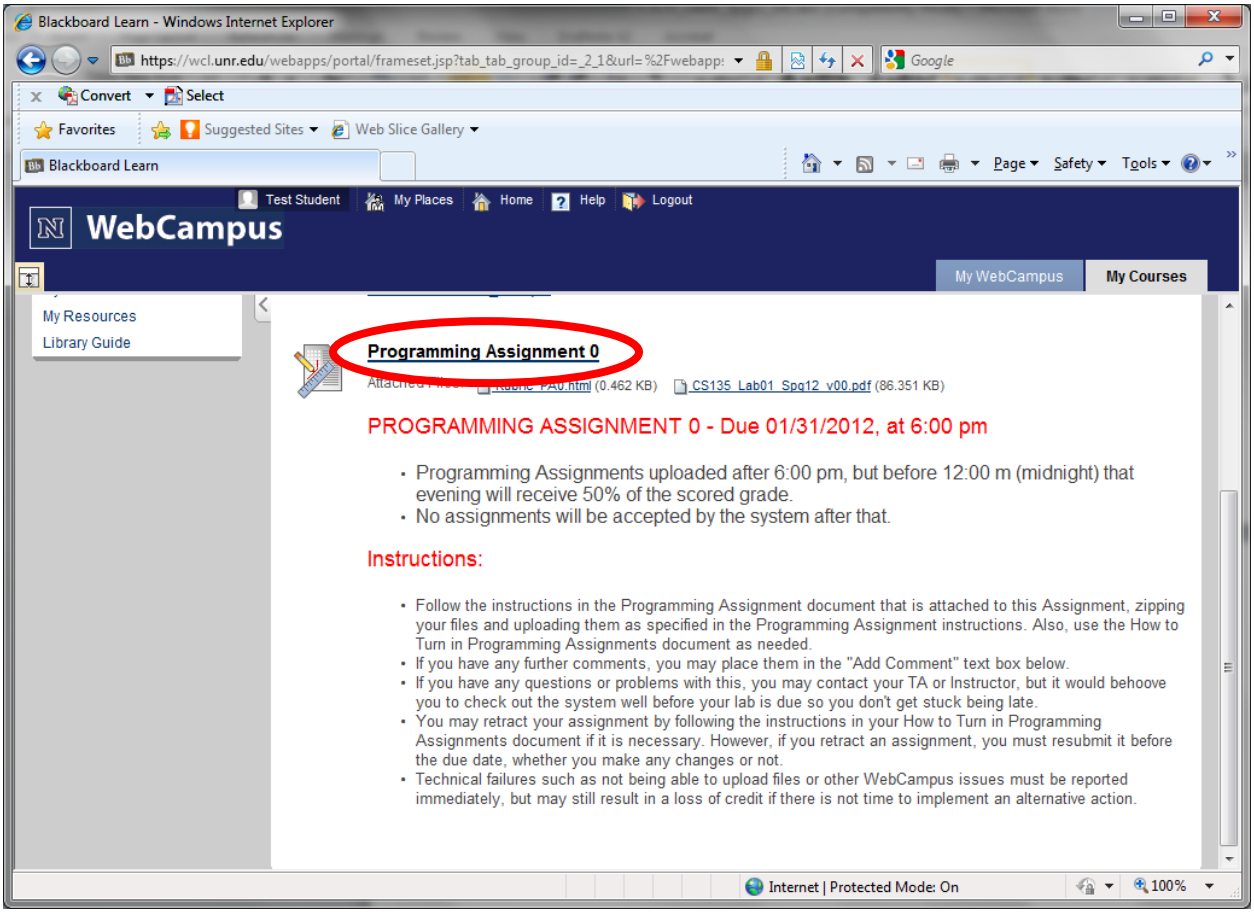
1. Collect the following files in one folder on your computer ("Week 1"):
    1. The Word file containing the following:
        1. There should be at least nine (9) screenshots for the program as specified above
        2. Remember to clearly annotate every displayed result
    2. The executable files:
        1. Rectangle_1.exe
        2. Triangle_1.exe
        3. Diamond_1.exe
        4. Rectangle_2.exe
        5. Triangle_2.exe
        6. Diamond_2.exe
        7. Rectangle_3.exe
        8. Triangle_3.exe
        9. Diamond_3.exe
    3. The source code files:
        1. Rectangle_1.cpp
        2. Triangle_1.cpp
        3. Diamond_1.cpp
        4. Rectangle_2.cpp
        5. Triangle_2.cpp
        6. Diamond_2.cpp
        7. Rectangle_3.cpp
        8. Triangle_3.cpp
        9. Diamond_3.cpp
2. Select all of these files, right click on them, and select “Send To”, then select “Compressed (zipped) Folder”.

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 PA0) as follows: "LeveringtonMichael_PA0" (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.

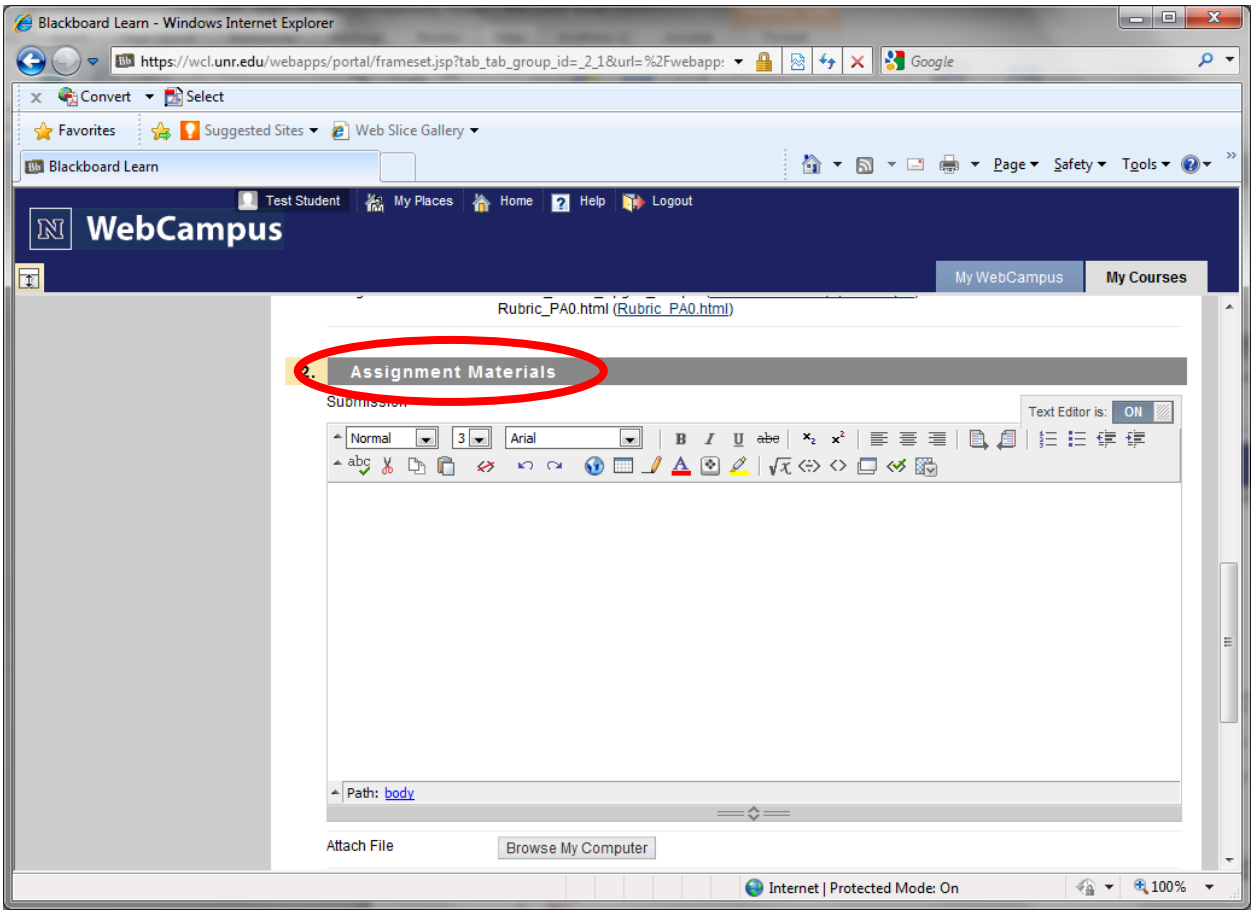
4. Go to WebCampus and go to the "Programming Assignment 0" in the "Week 1" folder. This is where you found the program materials initially. ***Make sure you are uploading your assignment to the correct place in WebCampus.***



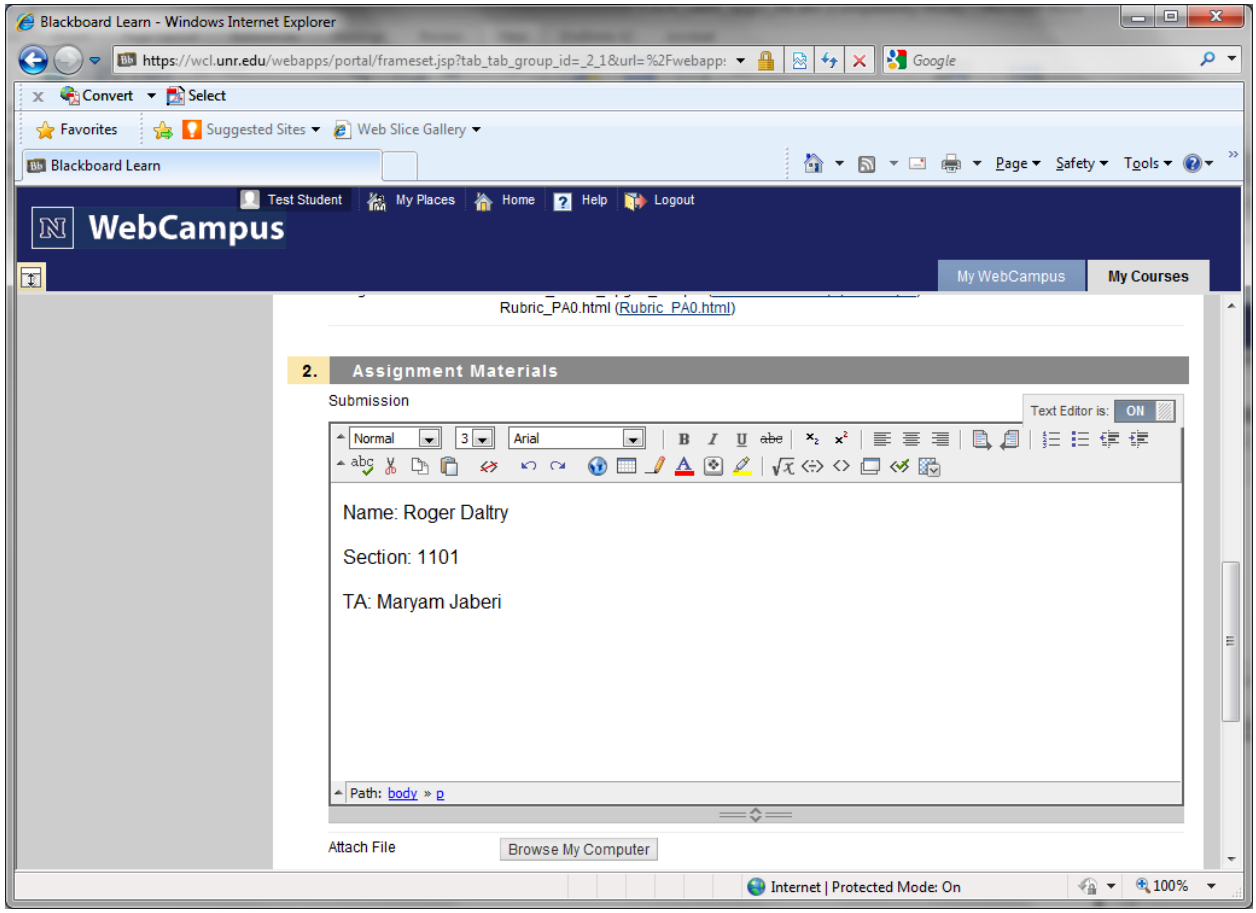
5. Select the "Programming Assignment 0" (the number will obviously change as the semester progresses).



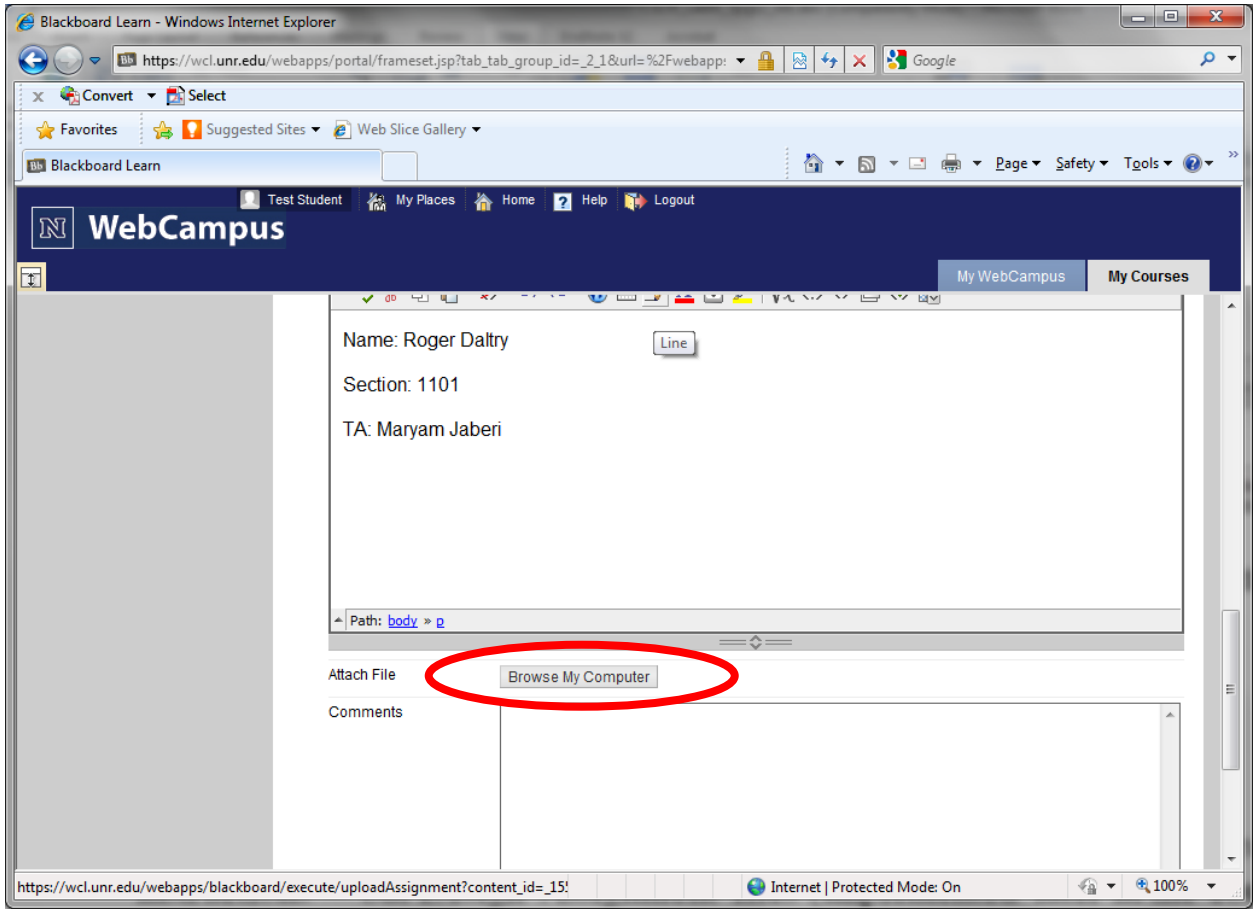
6. Go to the "Assignment Materials" section, as shown



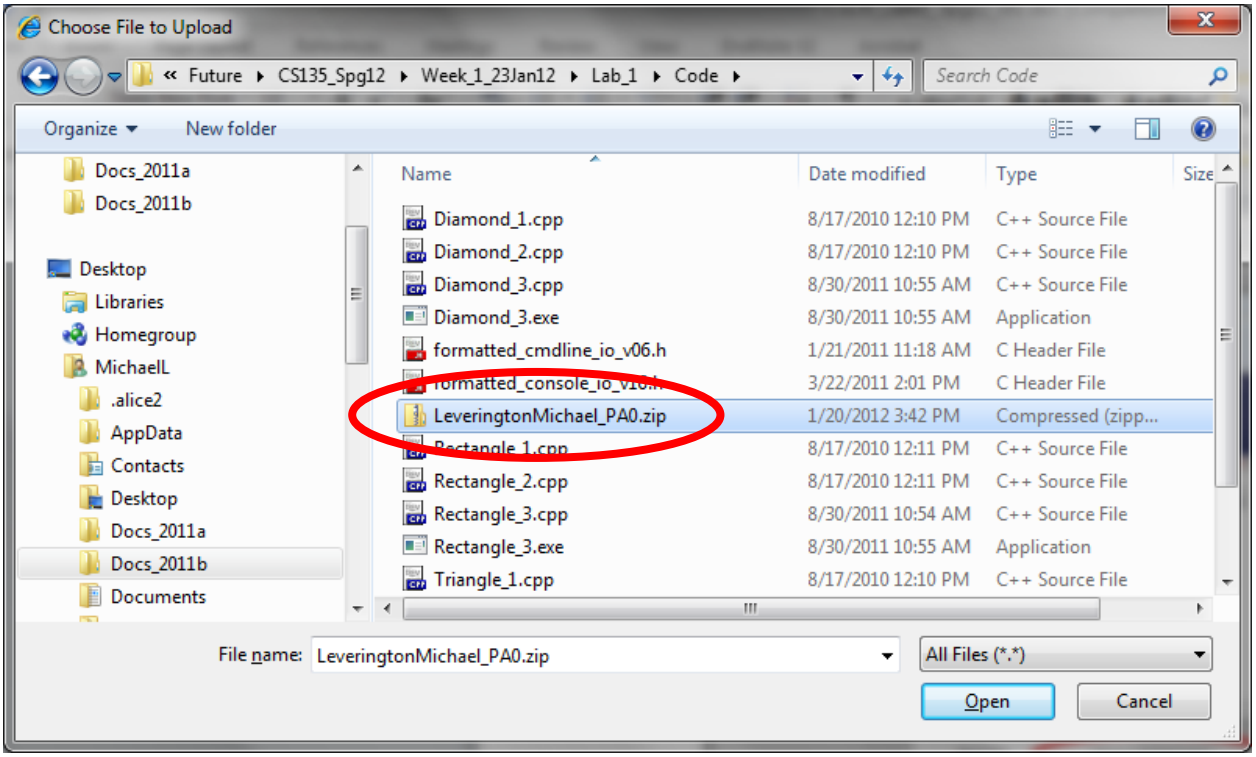
7. Enter the following information: 1) Your name, 2) your section number, and 3) your TA's name, as shown here



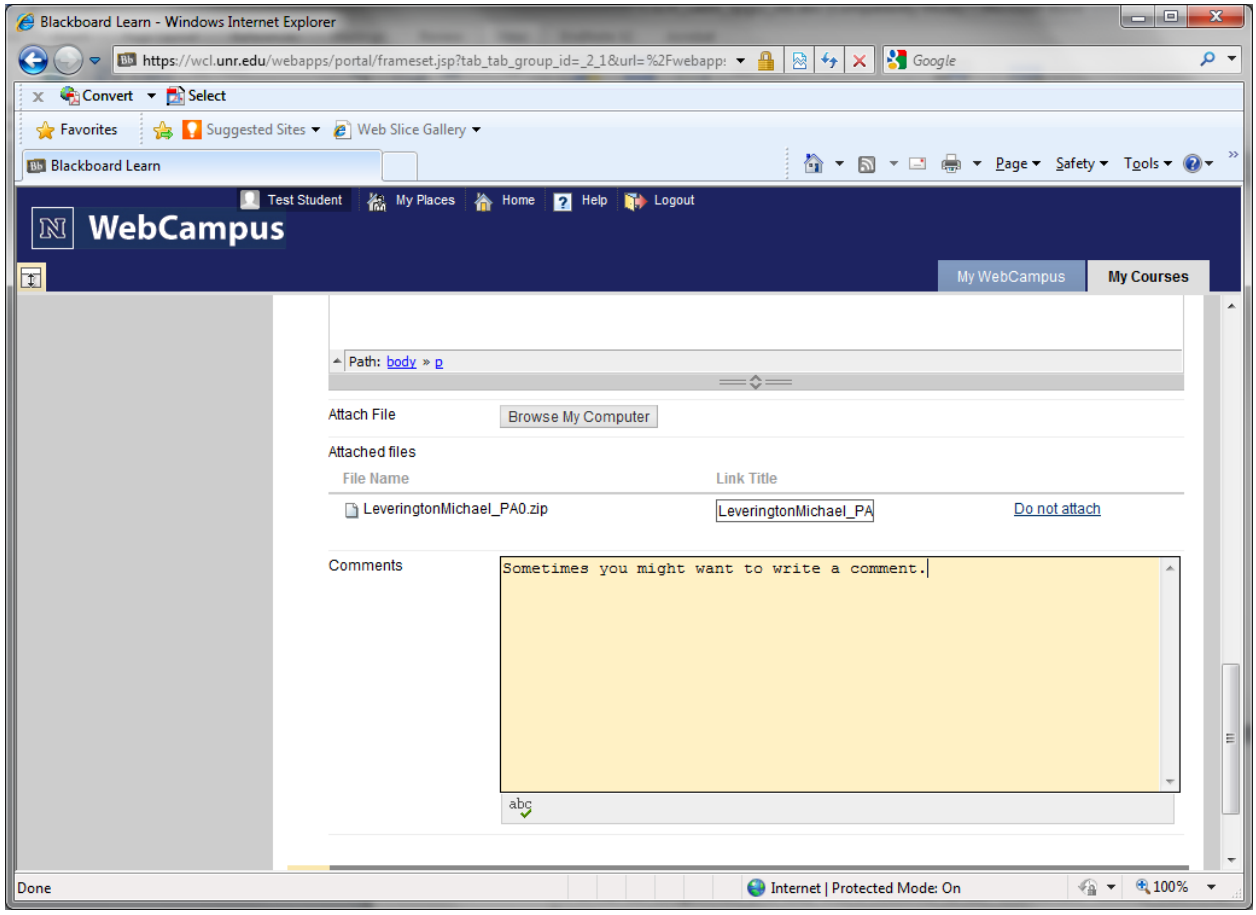
8. Then below the Submission box, check the "Browse My Computer" button to access your file.



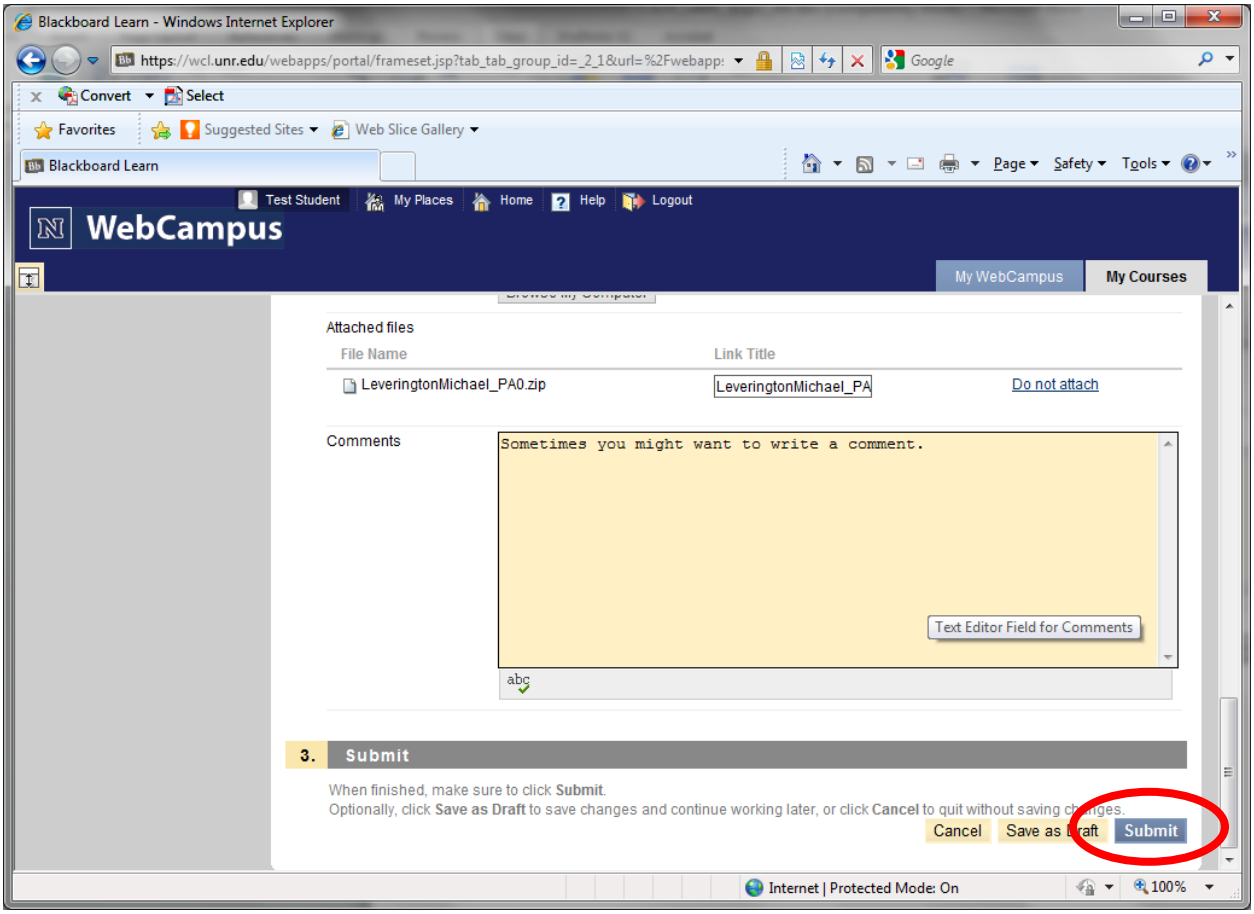
9. Select the file (or files when you are uploading Design Assignments later in the course), as shown



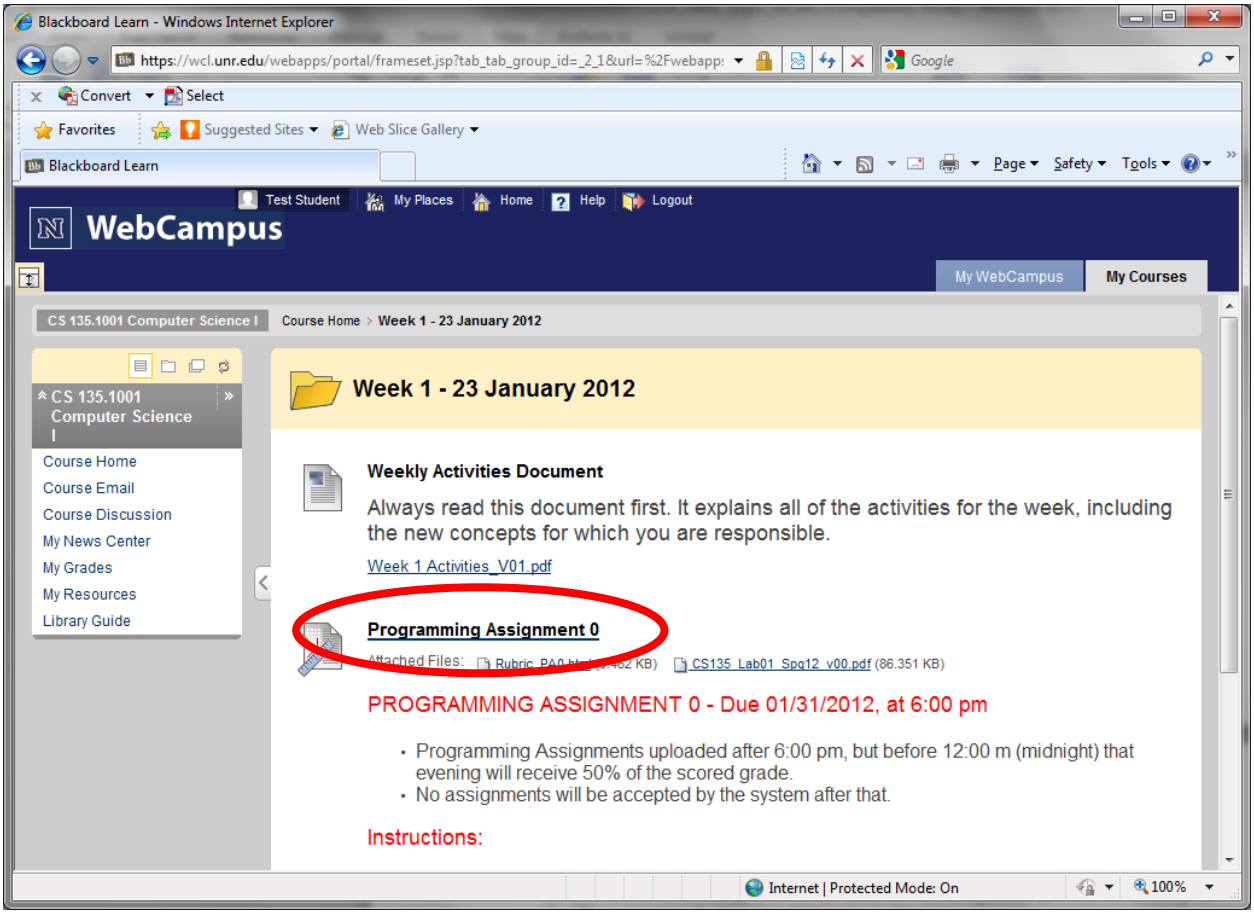
10. Once the file is uploaded, you will see it on the screen. If you need to make a note to your TA about the assignment, you may add a comment in the "Comments" section below, as shown. As a note, program files will commonly be zipped -- although you should always read the instructions. Design Assignment files (implemented later in the course) will commonly be uploaded as separate files.



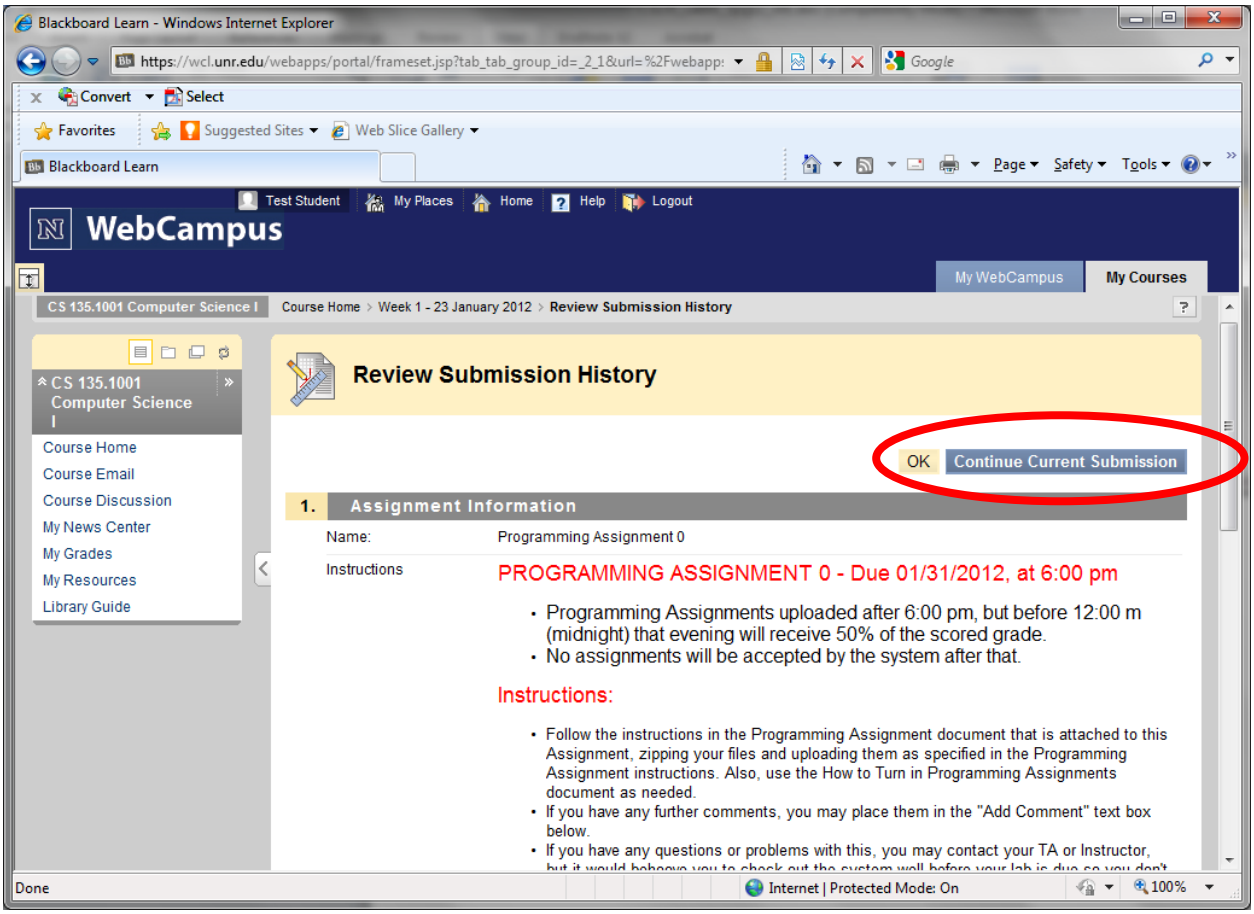
11. Once you have uploaded the file, you may submit it using the button in the lower right corner, as shown. As a note, you can save it as a draft if you are not sure about the completion, but if you forget to submit it, it will not be uploaded or graded.



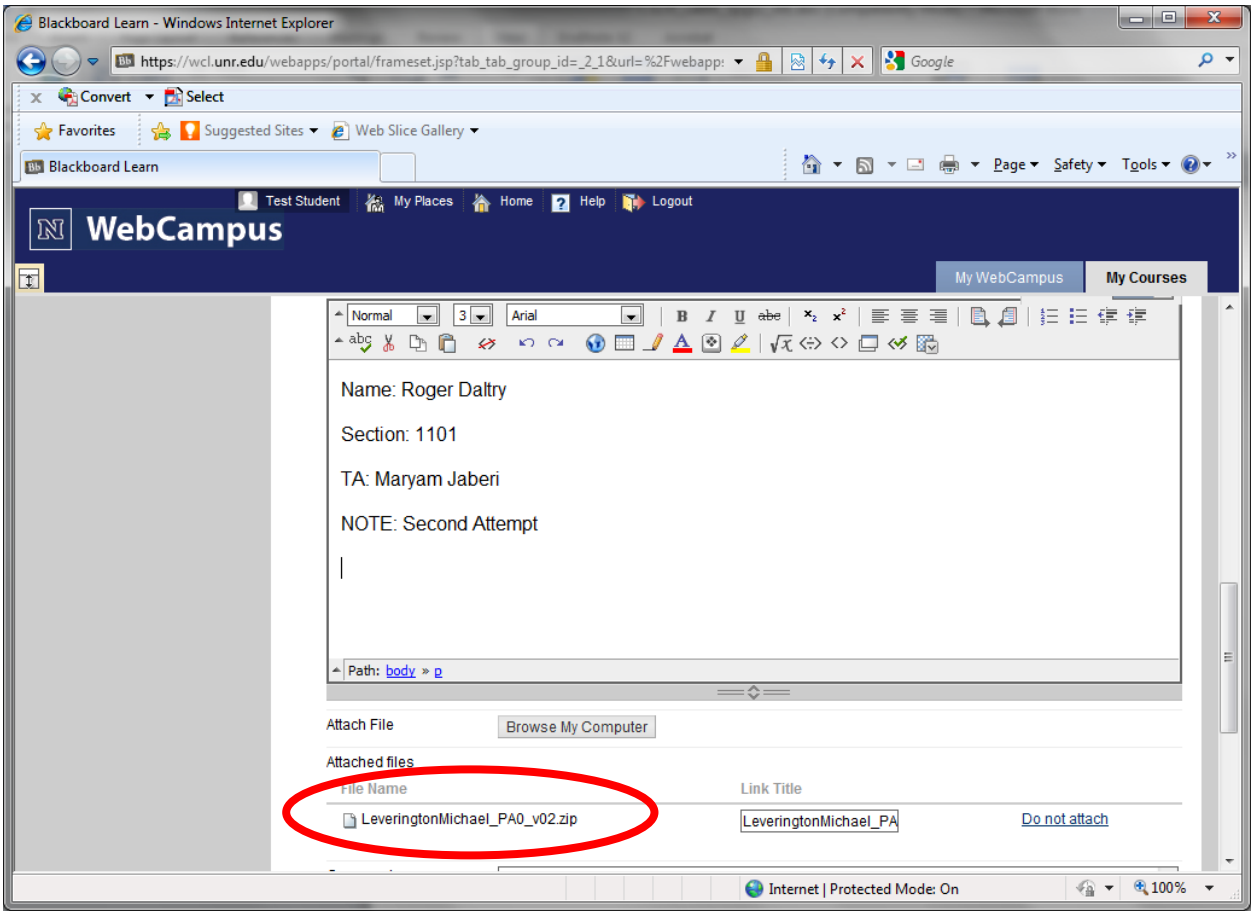
12. Once you have submitted the file, you cannot take it back. However, if you find that you need to correct something, you can go back into the assignment process by clicking on the "Programming Assignment 0" button again, as shown.



13. Select the "Continue Current Submission" button, as shown.



14. Repeat the uploading process, but rename your new file with a new version as shown. Also, show the "NOTE: Second Attempt" comment under the other information. If you do not make this note, your TA may not know which file to look for and the incorrect file may be reviewed.



15. If you have any questions about this process, you may contact the course Instructor(s) or TAs, but if you contact them at or near the deadline time, you may still lose credit if your assignment is late.