X414.20 Fundamentals of Software Development

Coding Lab 2

Please complete these lab steps only while in the Zoom meeting.

It is easiest to print these instructions so that you can refer to them during the lab. Your screen will be busy enough with both Zoom window and your Amazon Workspace window.

Task 1 – Install Amazon Workspaces client on your computer

1. Make sure you are on the computer you will be using for your classwork.
2. Find your **AWS Welcome Email** using your email program (such as Outlook or Mail) or email website.
3. Complete your user profile and download a WorkSpaces client using the link found in your AWS Welcome email. The link should be from unexlabs.awsapps.com. Do NOT go to the general public's AWS (Amazon Workspace) website.
4. Launch the client and carefully enter the registration code from the Welcome email.
5. Login with your newly created password. Your username is found in the Welcome email.
6. Once you see your AWS desktop, STOP AND WAIT FOR FURTHER INSTRUCTIONS.

Task 2 – Explore and tailor your AWS Environment

1. Resize your AWS Window as large as you can so that it is comfortable to see alongside your Zoom window. If you are on a small screen, you may need to flip back and forth from your Zoom window to your AWS windows. While you are working in class on Zoom, you should NOT set the AWS window to Full Screen (on the View menu), however, outside of class, this is the best way to work on it. (To escape from Full Screen, move your mouse point to the top of the screen and pause, and the menu will reappear).
2. Take a couple of minutes to explore the AWS environment. It is Windows 10. You should have the necessary icons for the three programs you will use in this course: Google Chrome, Dev C++ and Microsoft Visual Studio. These should be on your taskbar at the bottom of the AWS window, or on your Desktop, or both.
    a. Click on the Google Chrome icon to open it. Browse to the course's Canvas site (**my.uclaextension.edu**) and login to view the Canvas environment. Then minimize Google Chrome by clicking on its minimize button in the upper right corner of the Chrome window. **Do not minimize the AWS window by mistake.** The minimize button of Chrome is the lower one that appears in the corner.
    b. Click on the folder icon in the Taskbar to bring up the File Explorer. You will be working on the C: drive, which is NOT shown here. To see the C: drive, click once in the file specification area toward the top (just underneath Share and View in the menu, to the right of the words "Quick Access >". These words will then highlight. Then type c: and press ENTER. You should see the folders in the C: drive. This is one way of getting to the C: drive.

c. Close the File Explorer (click the X in its upper right corner). Now look for a C Drive icon on the desktop (let me know if it is NOT there). This will have been placed there for your convenience. Double-click on it to open it. You should now see the contents of the C: drive again.

d. Now you are going to create a **class** folder inside the C: drive. Click on **Home** on the menu in the File Explorer. This will display the *ribbon*, which gives you several more quick actions. Click the **New Folder** icon. A new folder is created and immediately type the word **class** and press ENTER. It should be all lower-case (this is important). If you need to correct it, hover over the name for second, click once (name highlights) and then either type the new name, or click once within the name to place the cursor there and then make the change using typing, backspace or delete.

e. Let's get you ready for programming by allowing you to always see full filenames with their extensions. Click on the **View** menu of the File Explorer. You'll see an item called **File name extensions**. Please click that box so that it is checked. To verify that it worked, let's view the C: drive again (remember how to do this?). Then open the Windows folder (double-click to open). Then scroll down past the folders to the individual files. You'll notice that all the files now display file extensions such as .exe, .log, .dll, .txt, .ini and more.

f. Let's finish by having you exit your AWS Workspace. The easiest way is to simply click the **X** in the upper right corner. BUT WAIT!!!! If you exit AWS this way, you could leave files and apps open, and although they stay open a while, AWS automatically shuts down the virtual machine (on the server at Amazon) after a while, and your work is not saved. So you should always save your work and close your applications before quickly exiting AWS. You can also go to the Windows button inside your AWS and choose the power icon and then Shut Down. This, however, will result in a longer boot up time if you wish to go back into the AWS.

Task 3 – Working in a Breakout Session (breakout session)

1. This and all subsequent tasks will be done in your breakout session with another student or two. Your instructor will assign you randomly to a breakout session with another studentor 2, and you will see and be able to communicate with your fellow students in Zoom. You will also be able to share your screen with the other students while in the breakout session. Before proceeding, make sure you are in the breakout session and can say hi to your partner(s) for this exercise. Be sure to Unmute and have your Video on.
2. Re-open your AWS. Share your AWS screen with your partner(s). Click on the Share Screen button in your Zoom meeting window. You will be given several choices, based on what applications you have open on your screen. If you see **Amazon Workspaces** listed as a choice, click on that and then on the **Share** button in the lower right corner. This way, only the AWS window will be shared. If this doesn't work, then you can choose to share your entire screen (it'll probably say either **Screen**, **Screen 1**, or **Screen 2**), but be careful since your personal non-class-related material, icons, photos, etc. may be visible.
3. Each of you should be able to share your screens, possibly even simultaneously. Take turns doing this so that you are comfortably with the size and placement of your windows. With only one physical screen it is difficult to display both AWS and Zoom, so you may end up talking

through your joint exercises and only sharing screens when necessary. During screen shares, you may want to see each other's video in miniature. If your video isn't already displayed, point to the top of the Screen Sharing window, and when the panel drops down, choose **Start Video**.

4. Signaling the Instructor. While in the breakout room, if you need the instructor's attention (either to ask a question or to check on work), click the **Ask for Help** button (question mark icon in a circle in a square). Please be patient; the instructor may be assisting in another breakout room. But when the instructor is ready, he will join your room and can view the shared screen and converse with you via audio and video. OK, now that you know how to do this, **try it out now and Ask for Help** so that the Instructor is signaled that you are ready to move to the next task.

Task 4 – Configuring Dev C++ (breakout session)

1. Click on the Dev C++ icon in the Taskbar. Dev C++ should open.
2. If you see any dialog boxes upon opening, **Ask for Help**.
3. You should see the Dev C++ screen. Size it accordingly within AWS.
4. Choose **Tools / Editor Options**. You will be placed in the **General** tab.
5. Make sure the following options are set accordingly:
    a. **Trim trailing spaces** should be CHECKED.
    b. **Cursor past EOF** should be UNCHECKED.
    c. **Cursor past EOL** should be UNCHECKED.
    d. **Use tab character** should be UNCHECKED.
    e. **Tab automatically indents** should be CHECKED.
    f. **Tab size** should be 3.
    g. Highlight **Current Line Enabled** should be CHECKED.
6. Click on the **Font** tab. Later, you may wish to adjust the font size for good readability by both you and your partner. This would be on the **Font** tab. Don't adjust it now, but reference this for later. If you wish to change it later, only change the **Size**, not the actual font.
7. Please click on **Line Numbers** in the same tab so that it is CHECKED.
8. Click on the **Color tab**. Down toward the bottom you can select a **Theme**. While I'd prefer that everyone use **Classic Plus** so that we are consistent, some of you love dark mode, so **Twilight** would be the one you might choose. Please don't use any other scheme and don't change anything else on the **Color** tab.
9. Click on the **Completion** tab. On the **Code Completion** subtab, UNCHECK **Enable code completion**. (At this early stage of coding, code completion really gets in your way.) Then click on the **Symbol Completion** subtab. **Enable Symbol Completion** should be CHECKED. But later if you decide this is getting on your nerves, you may uncheck it.
10. Click **OK** at the bottom to close the **Editor Options**.
11. Do an **Ask for Help** so that your instructor can check both of your environments before you proceed.

Task 5 – Type in this C Program, Save it, Compile and Run it (breakout session)

1. To start a new program, click **File** then **New** then **Source File**. This is a bit tricky with the mouse; it may be easier to simply hit **Ctrl+n**. (Mac users ***may*** have to use the **Command** (cloverleaf key) and **n**, if **Ctrl+n** doesn't work.) Or you can click the blank document icon on the icon bar.

2. Type the following program into the Dev C++ editor (please do NOT copy and paste it, you need to get used to typing). Make sure you substitute your name and your partner’s name in the first line.

```
// Coders: type your name here & type your partner’s name here
// Wow! My first program!

#include <stdio.h>  // Include this line in all of your programs

// Variable Declarations

   char name[10];   // This will hold the first name of the user

// *** Main Program ***

main(){
   printf("This is my first C program\n\n");
   printf("Please enter your first name: ");
   scanf("%s", &name);
   printf("\nThank you, %s\n", name);

   return 0;  // This line is in every program

}
```

3. Double-check the syntax and spelling **carefully**.
4. Save this as **myfirst.c** in your **class** folder:
    a. Click on the **File** menu.
    b. Click on **Save As…**.
    c. To see the C: drive, in the **File name** area, erase what is there (if anything), and type **C:** and press ENTER (or click **Save**). You should now see the folders in the C: drive.
    d. Double-click **class** folder to open it.
    e. Change the **Save as type** to C source files. (You should get in the habit of doing this in the future, or you won’t see your C files [extension of .c], only C++ files [extension of .cpp])
    f. In the **File name** area, type **myfirst.c** and then press **ENTER** or click **Save**.
5. Now let’s compile and run the program. Choose **Execute** on the menu and then **Compile & Run**. If you have function keys, you may also press the **F11** key. Some Windows laptop users will have to press **Fn+F11**.
    a. If there are no compilation errors, a new window will open up and the program will run in it. Type your name (no spaces) when asked and press **ENTER**. If everything works and looks good, then you are done! You can keep re-executing it with other values for fun. You’ll note that names with spaces won’t properly work. You’ll learn why this is next week. When ready, proceed to Task 7.
    b. If you have a compilation error, a window will NOT open up. Instead, you will be returned to the editor screen with a line highlighted. Your error is either on this line or just before. Read the first error message in the windows in the bottom to get a clue what the problem is. **Ask for Help** if you can’t figure it out. Make changes, save again (**File / Save**) and **Execute / Compile and Execute** again.

Task 7 – Submit the program in Canvas (breakout session)

1. Remember to save the program before exiting Dev C++. Choose **File** then **Save**. A quicker way to save is simply to press **Ctrl+s**. I like to do this periodically while I'm working, just to be careful.
2. Close Dev C++.
3. Re-open your Google Chrome. Be sure to choose Chrome from inside your AWS, not the one on your physical computer.
4. You should be in Canvas. If not, browse there and login. And make it your home page for ease of use throughout the course. (If you don't know how to do this, check with your partner or **Ask for Help**).
5. Go to **Modules**. Then **Module 2**. If it doesn't display automatically, click the wedge next to it to open things up. Click on **Coding Lab – M2 – Compile, Run and Submit a Program**.
6. Click the big **Submit Assignment** button.
7. Under **File Upload**, click **Choose File**.
8. In the File name box, type C: and press ENTER. Then double-click the class folder to open it.
9. Highlight the **myfirst.c** program, click it and then click **Open** (or you could simply double-click the file).
10. The file should appear next to the **Choose File** button.
11. You may add a comment to the **Comments…** area if you wish.
12. Then click **Submit Assignment**. The button will change to **Submitting** and then you will be returned to the Coding Lab screen. Assignment should be marked as Submitted! In the upper right corner.
13. Later, to check your score, click on **Grades** in the second-level menu on the left.
14. If I haven't graded your program yet, you will see an icon. Other, you will see a score. You can click on the name of the assignment to see my comments and to issue further comments or respond to mine.
15. Coding Labs will typically receive a score of 5 if you completed the lab and your submission was substantially correct. You may receive a score of 4 if there are some problems, and 3 if it is incomplete. You will receive a 0 if you did not participate in the lab. Ignore any "Total" scores that Canvas may show. They are meaningless.

Congratulations on completing your first Coding Lab!!! Please exit the breakout session to rejoin the entire class for final announcements.