# IT 230: Final Project Part I (ConsoleRegisterStudent) Coding Activity

**Name:** Grace Tay

**Date:** Nov 25, 2016

**Class:** IT 230

**Module:** 5

|  |  |
| --- | --- |
| **1.** | Insert a copy of your of the ZIP file of all of your Visual Studio project files here so that it can be loaded and run in another Visual Studio: |
| Insert here a copy of your \*.cs source code text you used here (copy and paste source code here, do **not** simply insert \*.cs files): |
| **2.** | Insert a screenshot here of the output that resulted from running your program, showing your last name as the first printed text to the screen:  *Screenshot of my code, comparable to the provided video:*    ../Desktop/Screen%20Shot%202016-11-25%20at%202.22.32%20PM.png  *After changing Convert.ToInt32 to TryParse and while loop* |
| **3.** | Explain the design of your program, the steps you took to complete it, and how you coded it:   1. I started my own program titled TayConsoleRegisterStudent as a console app and duplicated the provided code. 2. I built the solution to view the error list, which resulted in no code errors. 3. As I debugged, I began to see the errors. The biggest error I can see if that when you input your number, there is no statement saying “Registration Confirmed for course IT 145.” First thing I do is go and look for that statement in the code, where I see a switch statement. The only possible values that can be used as cases are returned by the int function ValidateChoice: -1, -2, -3, or -4; case 0 does not exist. Once I change case 0 to case -4, you can now register for classes. 4. I found another error when testing the registration confirmation statement, you can enroll in a class twice, which should not be permitted. The statement “You have already registered…” is in the switch(ValidateChoice(…)), so I look at the ValidateChoice function and what returns a -2 value. I changed && to ||, since this should be an “or” not “and” comparison. I also fixed a spelling error here, since it should be “registered”, not registerd. 5. While testing the last step, I tried to sign up for more than three classes. The program permitted me to do so, even though it should not be allowed. Again, I check switch(ValidateChoice(…)) for the “You cannot register…”, which is case -3, so I look at the ValidateChoice function and what returns a -3 value. When we reach the point that we have enrolled in 3 classes, our total credit will equal 9. The else if should check for totalCredit==9, not totalCredit > 9. Now, the program will not allow more than three classes. 6. I also notice that you can type in 8 and it does not tell me that the course is not recognized. This was an easy fix, simply changing the range in validateChoice to choice < 1 or choice > 7. 7. At this point, I check the video to walk through it and make sure my code responds identically. Everything works well! I included the screenshots at the top that reflect the video. Something I did want to change is the Convert.ToInt32, since if people type in a non-integer, like a letter “f”, it can crash the program. I already had existing code to handle this situation. This is the last screenshot seen above. This reduces the risk of the program closing without the user prompting the end of the program. |
| **4.** | Reflect on this experience and the lessons you learned from it:  I enjoy debugging assignments where there are no “code” errors in the error lists. It is important to consider human error and logic when debugging, since problems can arise from either code errors or just the thought behind it. It also challenges you a bit more to truly think about what the code is supposed to be, something I mentioned in our als0 assignment I believe. Another big part is thinking about your parameters that are passed into the function. For instance, a large chunk of errors could be attributed to the switch parameters. You need to look at what is being passed into the switch statement, which directs you to the ValidateChoice function. There, you have to look at the possible ints that can be returned, which are limited to -1, -2, -3, and -4. So, when you check the cases in the switch statement, you need to make sure only those numbers are reflected. I cannot imagine how difficult this will be when you have hundreds or thousands of lines of code and need to reference between functions (perhaps that is why you should test as you go and make sure you fix flaws before they become major problems!). Ideally, I would alter this code to have multiple classes. If this were an actual code project, you would likely have separate files for possible courses, checking for part of full time, perhaps a file writing system to write out all the classes a person registered for, etc. However, this was very efficient to just learn C# and learn more about switch statements and if/else statements. |