# An Auto-Feedback Program: if, if…else, switch

**The goal for this exercise** is to make sure that you understand how to use each of the three major conditional statements in C#: If, If…else, and switch statements.

Imagine that you're writing a plug-in for something like BlackBoard or ANGEL. Your plug-in will be told how many quiz questions a student has gotten correct on a 10-question quiz, and will print a message to the user giving appropriate feedback. Before you go creating the actual plug-in (which would involve learning a lot about the inner workings of BlackBoard), you decide to create a **prototype** – a simpler, easier version that still provides the core functionality, but doesn't have all the bells and whistles. In this case, the prototype will be a console application that gets the number of correct questions from the user, and prints out messages similar to the following:

|  |  |
| --- | --- |
| Number Correct | Feedback to send to the user |
| 0 | You need to talk to the professor immediately! |
| 1 | You need to study a lot more! |
| 2, 3, or 4 | If you keep working at this, you'll do better! |
| 5, 6 | You've gotten the hang of this, but with practice, I'm sure you can do even better! |
| 7, 8 | Nice job, but don't let yourself slack! |
| 9 | Very nice job – keep up the good work! |
| 10 | Perfect! |

If the user types in a number less than 0 you should print out a message saying that negative numbers aren't allowed. If the user types in a number greater than 10 you should print out a message telling them that their number is too high. If a user types in an invalid number (less than 0 or greater than 10), you don't have to do anything else (other than print out the appropriate message)

Type as much code as you want or need to. You should be able to solve this problem using only the statements and operators you’ve seen so far.

You should make sure that you can use all three of the following statements:

1. Using **if** statements
2. Using **if…else** statements
3. Using a **switch** statement (watch out for the break statements!)
(If the user types in an invalid number (too high or too low) then you can use an if/else to tell the user that, but you must generate the “Feedback to send to the user” message using only a switch statement (specifically including such cases as when the user gets 2, 3, or 4 correct!)

Avoid duplicating code – you should use the logical operators in order to minimize the amount of redundant code (especially for the **if**, and **if…else** statements). You should try to use 'fall-through' cases in your **switch** in order to avoid having a lot of the same, duplicate code for the **switch** exercise. (For more information on fall-through, see <http://www.dotnetperls.com/case> (but please do NOT switch on strings for this exercise – convert the user input to an integer, and switch on that, instead)

**What you need to do for this exercise:**  In the starter project, fill in the methods that Conditional\_Statements.RunExerise calls, namely Conditional\_Statements.UsingIf, Conditional\_Statements.UsingIfElse, and Conditional\_Statements.UsingSwitch.

|  |
| --- |
| In order to get full credit for this exercise, you need to create three separate versions of your code:1. One asks the user for input, and then uses **if** statements to generate the output. Put your code for this into the Conditional\_Statements.UsingIf method.
2. One asks the user for input, and then uses **if**…**else** statements to generate the output. Put your code for this into the Conditional\_Statements.UsingIfElse method.
3. One asks the user for input, and then uses a **switch** statement to generate the output. Put your code for this into the Conditional\_Statements.UsingSwitch method.
 |