Exercise – If Statements

First are quick review questions that you should write down the answers for in a document. The second part is practical exercises. Both are important for your learning and to help you retain the concepts.

Review Questions

1. Try and describe an **if statement**:

A decision making statement that guides a program to make decisions based on specific criteria.

2. What happens if the **condition** of an **if statement** is true?

The if statement will run certain code if the condition is true.

3. Describe how **else statements** work:

Else statements are blocks of code that will execute when the if statement condition is not true.

4. Describe how **else if statements** work:

You can have multiple else if statements after an if statement, the first if or else if statement condition that is true will run, the rest will be skipped until the program goes through the code again.

5. In a series of **if** and **else if statements,** if the first **if statement** condition is true, and there are 3 **else if statements** after it, how many of those **else if statements** will check their conditions? Why?

None. They get skipped if the first if statement is true.

6. Can you give an example of a nested **if statement?**

When an if statement is put inside of another if statement, the if statement inside another will run last.

7. What does the **<= Less Then Or Equal** relational operator do?

Checks to see if a the chosen variable is less than or equal to the value of a specified value.

8. What does the **!= Not Equal** relational operator do?

Checks to see if a chosen variable is not equal to a specified value.

Practical Exercise

Create a new C# console project in Visual Studio called **IfStatementsExercise.**
Add the following code to the **Main** function:

1. Create a **bool** called **playerAlive**, and initialise it's value to **true.**
2. Afterwards, create an **if statement** that checks to see if the **playerAlive** variable is **true.** If it is, print out some text telling the player they're still alive.
3. Create an **else statement** after it. Inside it, print text telling the player they are dead.
4. Create a new **int** called **invulnerabilityTimer,** and set it's value to something greater than 0**.**

5. On the next line, subtract some value from your **invulnerabilityTimer** variable
6. Create a new **if statement**, and for the condition, check to see if **invulnerabilityTimer** is **greater than 0.** If it is, print out text telling the player they are invulnerable.
7. Create an **else if statement** after the **if statement**, and set the condition to check if **invulnerabilityTimer** is **equal to 0.** If it is, print text telling the player they are now vulnerable to damage.
8. Create another **else if statement** after that, and set the condition to check if **invulnerabilityTimer** is **less than 0.** If it is, set the value of **invulnerabilityTimer** back to 0.
9. Finally, inside your **if statement** where you check to see if **invulnerabilityTimer** is **greater than 0,** nest another **if statement.** Set the condition to check if **playerAlive** has a value of **true.** If it does, subtract a value of 1 from **invulnerabilityTimer.**

Try changing how much gets subtracted from **invulnerabilityTimer** in step 5 to see how it affects the way your program runs.