

# Quest 7 (Key Quest)

200 EXP

CPSC121 SI

**Craig:** Welcome Back Programmer.

**Craig:** Today we will be covering if statements.

**Craig:** Before we get into if statements, Rex will teach you about relational operators.

**Rex:** Hello! A relational operator is used to compare values to determine relative order.

**Rex:** When you compare values using a relational operator, you will be given true or false.

**Rex:** ...

**Craig:** What's wrong Rex?

**Rex:** I know what relational operators are, but I have forgotten which were which.

**Craig:** (to you) Oh dear, can you help Rex match which relational operators mean what?

- |        |                             |
|--------|-----------------------------|
| ___ >  | a. Not equal to             |
| ___ <  | b. Greater than             |
| ___ == | c. Equal to                 |
| ___ != | d. Less than or equal to    |
| ___ <= | e. Greater than or equal to |
| ___ >= | f. Less than                |

**Craig:** Great! Now let's see if you can tell me if these conditions return true or false.

5 > 4 is \_\_\_\_\_

7 <= 3 is \_\_\_\_\_

If x = 6, y = 7 then:

x == 9 is \_\_\_\_\_

x != y is \_\_\_\_\_

y >= 7 is \_\_\_\_\_

**Craig:** Well done! We can get to the good stuff now.

**Craig:** If statements allow statements to be conditionally executed or skipped over. In other words, if a condition is met, the program will execute a block of code. If the condition is false, then the program is to skip that code block.

**Craig:** For example:

```
if( x >= 0 )
{
    cout << "x is zero or positive\n";
}
```

**Craig:** If x is greater than or equal to 0, then the program will execute the cout statement will be executed. If x is not greater than or equal to 0, then the cout statement will be skipped.

**Craig:** What happens when you want your program to do one thing if a condition is met, otherwise it does something else?

**Craig:** The answer is **else**. These are called if/else statements! Here's an example:

```
if( name == "Steven")
{
    cout << "Your name is Steven!\n";
}
else
{
    cout << "Hey, you're not Steven..." << endl;
}
cout << "I'm Freddy" << endl;
```

**Craig:** As you can see if the variable **name** holds the value "Steven" then you execute the first set of curly braces. Else you execute only the second set of curly braces. Also, regardless of if the condition is met or not, the program will display "I'm Freddy" next.

**Craig:** What happens if you want to check multiple conditions?

**Rex:** else if!

**Craig:** Exactly Rex. These are called if/else if statements. Shocker right? Here's an example:

```
if(grade >= 90)
{
    cout << "You got an A" << endl;
}
else if(grade >= 80)
{
    cout << "You got a B" << endl;
}
else
{
    cout << "You got a C or lower =" << endl;
}
```

**Craig:** In this example, the first condition is checked. If the condition is met, you only execute the first block of code. If the first condition is not met, but the second condition is, you execute the second block of code. If all the preceding conditions are not met, then the else code block is executed.

**Craig:** Now let's put our new knowledge to work. I need a program that asks for the names of three students, their grades, and also what is the class average. The program should then check if a student's grade is above the average, below the average, or is the average. Depending on which condition is met, display the student name and what their grade is in comparison to the provided class average.