

## Computer Science 3-4

### Problem Set #2: If Statements

We are now going to talk about one way C++ has to allow a program take an alternative action based on whether a certain condition is true or false. In an **if statement**, code is executed if the condition is true. If the condition is false, the program will skip ahead to the next section of code. Here is a table of ways we can compare values in C++ (which are slightly different than how we would write them out in a typical math class):

| Standard Algebraic Expression | C++ Expression | C++ Sample             | Meaning of C++ Condition        |
|-------------------------------|----------------|------------------------|---------------------------------|
| >                             | >              | <code>x &gt; y</code>  | x is greater than y             |
| <                             | <              | <code>x &lt; y</code>  | x is less than y                |
| ≥                             | >=             | <code>x &gt;= y</code> | x is greater than or equal to y |
| ≤                             | <=             | <code>x &lt;= y</code> | x is less than or equal to y    |
| =                             | ==             | <code>x == y</code>    | x is equal to y                 |
| ≠                             | !=             | <code>x != y</code>    | x is not equal to y             |

Here's an example of what a basic **if statement** looks like:

```
if (firstNumber == secondNumber)
{
    cout << "These numbers are the same!";
}
```

Note that we put the condition we want to test inside parentheses next to the “if” and the code we want to execute lives inside the curly braces (which lets us know that these statements belong to this particular “if”). Note that for readability, we indent the statements inside the braces one tab....this helps us keep track of which code goes with which functions in larger programs. It turns out that indentation is one of the more important standards we should follow when we code (along with commenting our code!).

This basic statement is great for when we only want to carry out some code when something is true...but more often than not, we'll want to do one thing if the statement is true and something else if it is false. To accomplish this, C++ let us look at the **if...else statement** which behaves exactly as it sounds. If the test condition is true, the code under the “if” statement is executed; if it is false, the statement associated with the “else” is run instead.

Here is a quick example:

```
if(grade >= 60)
{
    cout << "You pass!";
}
else
{
    cout << "You fail...\n";
    cout << "You must take this course again, sucker!";
}
```

When we want to test a series of conditions, we will want to "nest" if...else statements together, for example:

```
if (firstNumber == secondNumber)
{
    cout << "These numbers are the same!";
}

else if (firstNumber > secondNumber)
{
    cout << "The first number is larger!";
}

else
{
    cout << "The second number is larger!";
}
```

In this example, notice that we can put the **else if** statement together on the same line right after the curly brace. The final **else** sits by itself because it goes with the "if" statement directly above it. We won't have any more conditions to test (we checked to see if they were equal, then if the first number was bigger....if neither of those is true, then the second number must be larger, so we don't need another if statement). Remember that elegant programs try to take advantage of known information

One last note....last week we talked about the modulus operator (%) briefly. It is a function that lets us look at the remainder after we divide two numbers. So  $5 \% 4$  would give us 1 (since we have one left over after we divide 5 by 4). In math class, we write it as  $5 \bmod 4$ . This turns out to be very useful for some common programming tasks!

## The Problems!

1. Body Mass Index (BMI) calculator. Write a program that asks the user to input their weight in pounds and height in inches and then calculates and displays the user's BMI. Here is the BMI formula:

$$BMI = \frac{weight \times 703}{height^2}$$

The program should also indicate if the person is underweight, normal, overweight or obese as defined by this Health and Human Services Chart:

| HHS Classification | BMI Value           |
|--------------------|---------------------|
| Underweight        | Less than 18.5      |
| Normal             | Between 18.5 and 25 |
| Overweight         | Between 25 and 30   |
| Obese              | Greater than 30     |

2. Write a program that takes in the total points on an exam, the score the student got on the exam and outputs their letter grade. Use the grading scale for this class to decide which percentage scores give what letter grades.

3. Write a program that reads an integer and determines and prints whether it is odd or even (Hint: Use the modulus operator (%); any multiple of two will leave a remainder of zero when divided by 2).

4. Write a program that reads in two integers and determines and prints if the first is a multiple of the second (Hint: use the modulus operator).

5. **Challenge:** Write a program that inputs a five-digit integer, separates the integer into its digits and prints them separated by three spaces each. For instance, if the user enters 42339, the program should print out: 4 2 3 3 9. **Hint:** Think about some of the behavior we have seen with integers, division and the modulus operator.