

INTRO TO C++ FOR FRC: LESSON 2

Joel Kuehne

<https://github.com/JoeKueh/frc-cpp>

CONTENTS

1. Review: Programming as Math
2. Variables: Working with Data
3. Variables: Declaration and Initialization
4. Conditions: If Statements
5. A simple C++ Program
6. Homework: Learn C++ Syntax

- As stated in the last lesson, we can model mathematical functions in code.
 - We take functions that we want to replicate and turn them into steps.
 - We then take those steps and translate them into code.
- We have access to more features in code than just pure mathematical functions.
 - We can manipulate data (variables).
 - We can respond to events and check conditions (if statements).

- One of the most powerful tools a computer provides is the ability to manipulate data.
- Results of calculations can be saved for later in “variables”.
- In C++, the basic variable types are as follows
 - Integer (int): Stores integer data (...,-2,-1,0,1,2,...)
 - Float (float): Stores decimal data¹
 - Character (char): Stores a single character ('a','B','Z','x',...)
- You can define a variable by specifying it's type, then giving it a name.
 - See the next slide for an example.

¹Float stands for floating point. Think about how you might write a decimal number on paper, you write a sequence of numbers (e.g. 321543) and then choose where you want the decimal place to go (e.g. 32.1543 or 32154.3). The decimal point can go anywhere in the number, it is “floating”.

- You can declare variables like this.

```
int integer_variable;  
float decimal_variable;  
bool true_false_variable;  
char single_character;
```

- You can also give variables a default (initial) value like this.

```
int integer_variable = 15;           // Assign the integer 15  
float decimal_variable = 13.2;      // Assign the decimal 13.2  
bool true_false_variable = false;  // Assign the Boolean false  
char single_character = 'c';        // Assign the character 'c'.
```

- After a variable is declared, you can set its value using '='

```
integer_variable = 18;
```

Exercise 1: Basic Variable Usage

6 / 10

Give it a shot, what do you think the variable “a” will be at the end of this segment.

```
int a = 1;  
int b = 2;  
int c = 3;
```

```
a = b;  
b = c;  
a = b;
```

Exercise 1: Basic Variable Usage

7 / 10

If you said `a == 3` you'd be right.

- The next powerful tool for programmers is the ability to perform different actions based on the values of different variables.
- The simplest of these conditional blocks is the “if” statement.
- The “if” statement models the logic “if x is true then do y, else do z”.
- In C++, it looks like this.

```
int y = 0;  
int z = 0;  
bool x = true;  
if (x == true) {  
    y = 1;  
} else {  
    z = 1;  
}
```

- Note the difference between == and =. Double equals is used to check if two values are equal. Single equals is used to set the value of a variable.

A simple C++ Program

```
int main()
{
    // Set our initial conditions.
    int a = 3;
    int b = 5;
    int c = 8;

    // Check the condition a + b == c.
    if (a + b == c) {
        // This is how you print text in c++ (I know it's weird).
        std::cout << "They are equal!" << std::endl;
    } else {
        std::cout << "They are not equal!" << std::endl;
    }
}
```

- I hope these slides have given you a general idea of the tools that are at your disposal.
- The only way to get familiar with the programming concepts themselves, however, is to practice practice practice.
- Fortunately, there are some good tools for practicing C++ Syntax.
- Before next lesson, you all should look through
<https://www.geeksforgeeks.org/cpp/cpp-tutorial/>
- In particular, you should look through everything under “Basics” and the first two sections of “Function” (“Functions” and “Parameter Passing Techniques”).