C++ Programming Notes

*Based on w3schools*

**What is C++? (cpp)**

* Cross-platform language used to create high-performance applications. Developed by Bjarne Stroustrup as an extension to C.
* Gives high level of control over system resources and memory.

**Why use C++?**

* Found in today’s OS, GUI, and embedded systems.
* OOP language
* Adaptable to multiple platforms
* Similar to C, C#, and Java
* MAIN DIFFERENCE WITH C = Cpp supports classes and objects

**C++ Syntax**

To understand the structure of a Cpp program, we can use the Hello World example

|  |
| --- |
| #include <iostream> using namespace std;  int main() {   cout << "Hello World!";   return 0; } |

Breakdown of the code:

* Line 1: header file library, lets us work with I/O, add functionality to programs.
* Line 2: allows us to use names from the standard library.
* Line 4: main function of the program.
* Line 5: object used to print text.
* Line 6: ends the function.

Note: always end wih “;”

We can omit namespace by adding “**std::”** to the first line of the main function.

**C++ Output**

*cout* object with << operator is used to print text. Note that this doesn’t insert a new line at the end of the output.

* To insert a new line, use \n character – we can use “endl” the same way
* This is also called an **escape sequence**

A screenshot of a phone

Description automatically generated

**C++ Comments**

Used to explain the code and make it more readable. It can be used to prevent execution. Can be single lined (uses “//”, ignores text between // and end of line) or multi-lined (starts with “/\*” and ends with “\*/”, ignores text between start and end).

**C++ Variables**

Variables are containers for storing data values. There are different types of variables.

* int = stores integers (whole numbers), ex 123
* double = stores floating numbers (decimals), ex 19.9
* char = stores single characters, surrounded by single quotes, ex ‘a’
* string = stores text, surrounded by double quotes, ex “Hello World”
* bool = stores true or false

DECLARE VARIABLE

|  |
| --- |
| type variableName = value;  // example  int myNum = 15; |

* type = one of Cpp types
* variableName = name of variable
* equal sign = assign values to variable

It’s possible to declare a variable without assigning the value (assign later). You can overwrite the value to an existing variable as well.

|  |
| --- |
| int myNum; myNum = 15;  int myNum = 15;  // myNum is 15  myNum = 10;  // Now myNum is 10 |

Use cout to display variables.

|  |
| --- |
| cout << "I am " << myAge << " years old."; |

To declare many variables of the **same type**, use a comma-separated list. You can also assign the **same value** to multiple variables.

|  |
| --- |
| int x = 5, y = 6, z = 50;  int x, y, z;  x = y = z = 50; |

IDENTIFIERS

Variables must be identified with unique names or identifiers. It is recommended to use descriptive names to create understandable and maintainable code.

|  |
| --- |
| int minutesPerHour = 60; |

Rule of Thumb to naming variables:

* May contain letters, digits, and underscores.
* MUST begin with letters or underscoe.
* Names are case-sensitive.
* May NOT contain whitespace or special characters.
* Reserved Cpp words can’t be used as names.

CONSTANTS

To disallow the changing of variables, use *const* – which means unchangeable and read-only variable. Use for common, well-known values like pi.

|  |
| --- |
| **const** int myNum = 15;  // myNum will always be 15 |

* **Must** be assigned with a value

**C++ User Input**

Use *cin* to get input. This reads data from the keyboard with >> operator. Simple case:

|  |
| --- |
| int x;  cout << "Type a number: "; // Type a number and press enter  cin >> x; // Get user input from the keyboard  cout << "Your number is: " << x; // Display the input value |