

Basic C++ Programming

Basic Program Start Coding

Libraries

Main Libraries

`#include <iostream>` : Calls the library within Dev C++ to control the proper Input/Output of C++ programming language. Used as the first line of any program.

`using namespace std;` : Calls the standard viewing space. Used as the second line of code in any program

Basic Codes

`int main()` : Initiates the main body of the program as an integer (works with return 0 (turns off program); return 1 would be infinite program).

`{ }` : Braces represents the start and end of a program. All programming code must be between the first and last brace

`;` : ends a line of code

`\n` : Written after a set of text in “ ” to enter down to the next line

`//` : Act as comments for the program. Any single line of code after the two backslashes will be omitted from compiling

`/* */` : Multiple lines of code can be commented out at one time

`endl;` : drops the cursor down to the next line on the screen

`return 0;` : Ends all programs. Sends back a value of 0 (Null) back to the operating system telling it the program has been completed. May Return other values if the program is a subroutine to a Main Program.

Input/Output Code

`cout << “ ”` : Sends whatever is listed between quotes to the screen or what the value of a variable is equal to.

`cin >>` : Has the user enter some form of data

Variable Types

`int`: Integer Value or Whole numbers. Can also be declared as a short or long value

`float`: Allows for real number values to be used. Allows use of decimal values. Can also be declared as float or double

`char`: Single character entry

`string`: Word character entry

`bool`: Logical entry. Typically True/False Statements

Mathematical Symbols

Standard Symbols +, -, *, /

Mathematical Comparison Symbol	Description
==	= "equals"
!=	Not Equals
>	> Greater Than
<	< Less Than
>=	>= Greater than or Equal to
<=	<= Less Than or Equal to
&&	&& And Statement
	And/Or Statement (Keystroke shift+backslash key)

Statements

<pre>If (condition) { statements }</pre>	Carries out the Statements of Condition if True
<pre>If (condition) { statement } else{ statements }</pre>	Checks to see if Condition is True, if so then carries out the Statement below and after skips the else. If Condition is False then carries out the Else Statements.
<pre>If (condition) { statement } else if (Condition) { statements } Else{ Statements }</pre>	Checks to see if Condition is True, if so then carries out the Statement below and after skips the if/else and else commands. If the first Condition is False then checks the second condition: If true then carries out the statement if False then carries out the final Else Statements.

Loops

<pre>While (conditions, iterator variable, variable decrement or increment) { statements }</pre>	Goes through the loop while the Conditional Statement is True. Once False the Program will fall out of the loop continuing on with the program.
<pre>For (conditions, iterator variable, variable decrement or increment) { Statements }</pre>	Goes through the loop while the Conditional Statement is True. Once False the Program will fall out of the loop continuing on with the program.
<pre>Do { Statements } while (conditions, iterator variable, variable decrement or increment);</pre>	Goes through the loop while the Conditional Statement is True. Once False the Program will fall out of the loop continuing on with the program.