**Integer Numerals.**

|  |  |  |
| --- | --- | --- |
| 1 2 3 | 1776  707  -273 |  |

These are numerical constants that identify integer values. Notice that they are not enclosed in quotes or any other special character; they are a simple succession of digits representing a whole number in decimal base; for example, 1776 always represents the value *one thousand seven hundred seventy-six*.

In addition to decimal numbers (those that most of us use every day), C++ allows the use of octal numbers (base 8) and hexadecimal numbers (base 16) as literal constants. For octal literals, the digits are preceded with a 0 (zero) character. And for hexadecimal, they are preceded by the characters 0x (zero, x). For example, the following literal constants are all equivalent to each other:

|  |  |  |
| --- | --- | --- |
| 1 2 3 | 75 // decimal  0113 // octal  0x4b // hexadecimal |  |

All of these represent the same number: 75 (seventy-five) expressed as a base-10 numeral, octal numeral and hexadecimal numeral, respectively.

These literal constants have a type, just like variables. By default, integer literals are of type *int*. However, certain suffixes may be appended to an integer literal to specify a different integer type:

|  |  |
| --- | --- |
| **Suffix** | **Type modifier** |
| u *or* U | unsigned |
| l *or* L | long |
| ll *or* LL | long long |

Unsigned may be combined with any of the other two in any order to form *unsigned* *long* or *unsigned long long*.  
For example:

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 5 | 75 // int  75u // unsigned int  75l // long  75ul // unsigned long  75lu // unsigned long |  |

In all the cases above, the suffix can be specified using either upper or lowercase letters.