Table 5-1. Variable types of C++.<sup>a</sup>

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
<th>Storage capacity</th>
</tr>
</thead>
</table>
| int         | 4 bytes 32 bits | -2,147,483,648 ... 2,147,483,647  
-80000000H ... 7FFFFFFFH |
| unsigned int| 4 bytes 32 bits | 0 ... 4,294,967,295  
0H ... FFFFFFFFH |
| short       | 2 bytes 16 bits  | -32,768 ... 32,767  
-8000H ... 7FFFH |
| unsigned short | 2 bytes 16 bits  | 0 ... 65535  
0H ... FFFH |
| long        | 4 bytes 32 bits | -2,147,483,648 ... 2,147,483,647  
-80000000H ... 7FFFFFFFH |
| unsigned long | 4 bytes 32 bits  | 0 ... 4,294,967,295  
0H ... FFFFFFFFH |
| char        | 1 byte 8 bits  | -128 ... 127  
-80H ... 7FH |
| unsigned char | 1 byte 8 bits  | 0 ... 255  
0H ... FFH |
| bool        | 1 byte 8 bits  | 0 (false) or 1 (true) |
| float       | 4 bytes 32 bits | Precision: 7 decimal digits  
Exponent range: -38 ... 38 |
| double      | 8 bytes 64 bits | Precision: 15 decimal digits  
Exponent range: -308 ... 308 |
| long double | 10 bytes 80 bits | Precision: 18 decimal digits  
Exponent range: -4932 ... 4932 |
| memory address<sup>b</sup> | 4 bytes 32 bits | Sufficient for addressing up to 4,294,967,296 bytes (4 gigabytes) of main memory. |

---

<sup>a</sup> The sizes and storage capacities presented in this table are those that we assume in this book. Later on in this book we assume that a variable of type int is a 32-bit variable, short is a 16-bit variable, and char is a 8-bit variable, etc. We shall also assume that a memory address is a 32-bit value. By making these assumptions it is easier to speak about the programs. The C++ language standard does not define exact sizes for different variable types. Although the sizes that we assume in this book are very common in modern C++ compilers, there are still C++ compilers in which variables of type int are 16-bit variables. In the future there might be C++ compilers in which an int variable has 64 bits.

<sup>b</sup> "memory address" is not a variable type in C++. This row describes how much memory is used for pointers which store memory addresses in C++ programs.

This is a sample page from Kari Laitinen’s book "A Natural Introduction to Computer Programming with C++"
More information can be found at www.naturalprogramming.com