2.18 Unsigned

Sometimes a programmer knows that a variable's numbers will always be positive (0 or greater), such as when the variable stores a person's age or weight. The programmer can prepend the word "unsigned" to inform the compiler that the integers will always be positive. Because the integer's sign needs not be stored, the integer range reaches slightly higher numbers, as follows:

Table 2.18.1: Unsigned integer data types.

Declaration	Size	Supported number range	Standard-defined minimum size
unsigned char myVar;	8 bits	0 to 255	8 bits
unsigned short myVar;	16 bits	0 to 65,535	16 bits
unsigned long myVar;	32 bits	0 to 4,294,967,295	32 bits
unsigned long long myVar;	64 bits	0 to 184,467,440,737,095,551,615	64 bits
unsigned int myVar;	32 bits	0 to 4,294,967,295	*16 bits*

Feedback?

Signed numbers use the leftmost bit to store a number's sign, and thus the largest magnitude of a positive or negative integer is half the magnitude for an unsigned integer. Signed numbers actually use a more complicated representation called two's complement, but that's beyond our scope.

The following example demonstrates the use of unsigned long and unsigned long long variables to convert memory size.

Figure 2.18.1: Unsigned variables example: Memory size converter.

```
#include <iostream>
using namespace std;

int main() {
   unsigned long memSizeGB;
   unsigned long long memSizeBytes;
   unsigned long long memSizeBits;

   cout << "Enter memory size in GBs: ";
   cin >> memSizeGB;

   // 1 Gbyte = 1024 Mbytes, 1 Mbyte = 1024 Kbytes, 1 Kbyte = 1024 bytes
   memSizeBytes = memSizeGB * (1024 * 1024 * 1024);
   // 1 byte = 8 bits
   memSizeBits = memSizeBytes * 8;

   cout << "Memory size in bytes: " << memSizeBytes << endl;
   cout << "Memory size in bits: " << memSizeBits << endl;

   return 0;
}
```

```
Enter memory size in GBs: 1
Memory size in bytes:
1073741824
Memory size in bits:
8589934592

...

Enter memory size in GBs: 4
Memory size in bytes:
4294967296
Memory size in bits:
34359738368
```

Feedback?

PARTICIPATION ACTIVITY 2.18.1: Unsigned variables.

1) Declare a 64-bit unsigned integer variable numMolecules.

Check Show answer

Answer

unsigned long long numMolecules;

An unsigned long is 32 bits. An unsigned long long is 64 bits.

2) Declare a 16-bit unsigned integer variable named numAtoms.

Check Show answer

Answer

unsigned short numAtoms;

An unsigned short is 16 bits.

3) Initialize numAtoms to the smallest valid unsigned value.

```
unsigned short numAtoms
= ;
```

Check Show answer

Answer

0

The smallest value an unsigned variable can be assigned is 0.

Feedback?