Random numbers in C++

The built-in random number generator in C++ (called `rand()`) generates a random integer between 0 and a very large number denoted by the built-in constant RAND_MAX. Unlike Python, there is no built-in function to generate a random number between a specific upper and lower bound. Therefore, the most common way to accomplish this is to take the remainder of the return value of `rand()` with the upper bound and add a constant as follows:

```
int v1 = rand() % 100;          // v1 is in the range 0 to 99
int v2 = rand() % 100 + 1;      // v2 is in the range 1 to 100
int v3 = rand() % 30 + 1985;    // v3 is in the range 1985-2014
```

A further complication arises because `rand()` always generates the same sequence of random numbers every time you run your program (unlike Python). (This is actually a useful feature in situations when you need to debug a program that uses random numbers and you want the program to act exactly the same way each time you run it.) To get around this, put the following line of code at the beginning of `main()`:

```
srand(time(0))
```

This "seeds" the random number generator with a new value (based on the current time) each time your program runs, so you get new random numbers each time.

These functions live in the `<cstdlib>` and `<ctime>` libraries, so you should `#include` both of those files at the top of your code when you generate random numbers.

Example:

```
#include <cstdlib>
#include <iostream>
#include <ctime>

using namespace std;

int main()
{
    srand(time(0)); // use current time as seed for random generator
    int random_variable = rand() % 100 + 1;

}
```

Note --- even if you use rand() from inside a function, only put the srand(time(0)) call once at the top of main().