The Structure of `switch` statements

The `switch` statement is similar to an `if-else ladder`. It is used to choose from among many courses of action depending on the value of a particular variable (which must be an integer or a character). The syntax is as follows:

```
switch(switchvariable)
{
  case firstvalue:  statement(s);
                    break;
  case secondvalue: statement(s);
                    break;
  [more cases optional]
  [default:         statement(s);]
}
```

Because this syntax may appear rather opaque, we present the following example:

```
int usernumber, option=4;
cout << "Enter your user number: ";
cin >> usernumber;

switch (usernumber)
{
  case 0:   cout << "You are the null user!" << endl;
            break;
  case 42:  if(option==1)
              cout << "You are all-knowing!" << endl;
            else
              cout << "You are almost all-knowing!" << endl;
            break;
  case 999: cout << "You are the superuser!" << endl;
            break;
  default:  cout << "You are some random user." << endl;
}
```

The `default` clause is optional. The `break` statement is important—without it, the appropriate statements *and* all statements that follow it are executed. In the example above, if there were no `breaks`, the "null user" would see all four messages. All code written after the case is listed will be executed up to the break, so loops, if_else etc. may be used.