Ekaba Bisong
Programming in C++
University of Calabar

Lesson Note #33
June 09, 2015

Adapted from C++ Tutorial at http://www.cplusplus.com/doc/tutorial/control/
and C++ How To Program edited for our own purposes

Introducing enum

Enumerations create new data types to contain something different that is not limited to the values fundamental data types may take. Its form is the following:

```
enum enumeration_name {
  value1,
  value2,
  value3,
  .
  .
} object_names;
```

For example, we could create a new type of variable called **color** to store colors with the following declaration:

```
enum colors_t {black, blue, green, cyan, red, purple, yellow, white};
```

Notice that we do not include any fundamental data type in the declaration. To say it somehow, we have created a whole new data type from scratch without basing it on any other existing type. The possible values that variables of this new type **colors_t** may take are the new constant values included within braces. For example, once the **colors_t** enumeration is declared the following expressions will be valid:

```
colors_t mycolor;

mycolor = blue;

if (mycolor == green)
{
    mycolor = red;
}
```

Enumerations are type compatible with numeric variables, so their constants are always assigned an integer numerical value internally. If it is not specified, the

integer value equivalent to the first possible value is equivalent to **0** and the following ones follow a +1 progression. Thus, in our data type **colors_t** that we have defined above, **black** would be equivalent to **0, blue** would be equivalent to **1, green** to **2**, and so on.

We can explicitly specify an integer value for any of the constant values that our enumerated type can take. If the constant value that follows it is not given an integer value, it is automatically assumed the same value as the previous one plus one. For example:

```
enum months_t { january=1, february, march, april,
                may, june, july, august,
                september, october, november, december} y2k;
```

In this case, variable **y2k** of enumerated type **months_t** can contain any of the 12 possible values that go from **january** to **december** and that are equivalent to values between **1** and **12** (not between **0** and **11**, since we have made **january** equal to **1**).