Polymorphism

The values shown in this box are calculated if the functions Volume are declared to be ***virtual***, which means that the references are resolved at ***run*** time. Notice the run time correctly calculates the values of CBox and CGlassbox.

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

class CBox
{
 public :
 void ShowVolume() const
  { cout << endl
         << "CBox1 usuable volume is "<<Volume();
  }
  // Function to calculate the volume of a CBox object
  virtual double Volume() const
  {
    return m_Length*m_Width*m_Height; }

  //Constructor
  CBox(double lv=1.0, double wv=1.0, double hv=1.0)
                    :m_Length(lv), m_Width(wv), m_Height(hv) {}

   protected :
     double m_Length;
     double m_Width;
     double m_Height;

};
```

This example comes from Ivor Horton Visual C++2010 WROX publishers pages 572-577.

```
#include "box.h"

class CGlassBox: public CBox
{
 public :
 void ShowVolume() const
  { cout << endl
         << "CBox 2 usuable volume is "<<Volume();
  }
  // Function to calculate the volume of a CBox object
  virtual double Volume() const
  { return 0.85*m_Length*m_Width*m_Height; }

  //Constructor
  CGlassBox(double lv, double wv, double hv) : CBox(lv, wv, hv){}


};
```

This function is declared to be const, since there is no valid reason to change the values of the class. It is refered to as an ***accessor*** or ***getter*** function

```
1   using namespace std;
2   #include <iostream>
3   #include "glassbox.h"
4
5   int main ()
6   {
7     CBox myBox(10.0, 10.0, 10.0);
8     CGlassBox myGlassBox (10.0, 10.0, 10.0);
9
10    CBox *pBox=NULL;
11    pBox=&myBox;
12    pBox->ShowVolume();
13    pBox=&myGlassBox;
14    pBox->ShowVolume();
15    cout <<endl;
16  return 0;
17  }
```

Functions which change the values of the variables in a class should nots declared to be const. These are refered to as an ***mutator*** or ***setter*** functions. Think of Tater in cars.

```
CBox usuable volume is 1000
CBox usuable volume is 850
```

```
CBox usuable volume is 1000
CBox usuable volume is 1000
```

The values shown in this box are calculated if the functions Volume are ***not declared to be virtual***, which means that the references are resolved at ***compile*** time. The compiler resolves the reference in the class CBox rather CGlassBox. This results in an incorrect value for myGlassBox since it inherits it values from CBox and both classes have a function called Volume.