An Introduction to C++ Classes

C++ classes are similar to structs. For example, consider the following struct.

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

struct Fraction
{
  long Numer;     //holds numerator of the Fraction
  long Denom;     //holds denominator
};

void main()
{
  Fraction F;

  cout << "Enter the numerator and denominator of a fraction\n";
  cin >> F.Numer >> F.Denom;

  cout << "The fraction is : " << F.Numer << '/' << F.Denom << endl;
}
```

If we replace the word "*struct*" with the word "*class*" the program behavior would be identical, except for one thing: the program would not compile! The compiler would display an error message something like this:

```
"Fraction::Numer is not accessible"
```

The explanation: fields of a class are "private" by default and cannot be accessed in main. The syntax error goes away if we make a small change, adding "public:".

```
class Fraction
{
  public:         //Now fields below are accessible in main
  long Numer;     //holds numerator of the Fraction
  long Denom;     //holds denominator
};
```

At first glance, this might seem strange, but there is a powerful motive behind it: there needs to be a mechanism for hiding (limiting access) data. We'll explore this idea more carefully later. Let's assume for now that we want to hide the **class members** *Numer* and *Denom* and see how the struct based program could be written with classes. The trick is to write **member functions** that perform any needed actions. Member functions can be defined inside the class (i.e. before the semicolon that marks the end of the class) or outside, in which case the **scope resolution operator** must be used to tell the compiler that they are part of the class. Member functions have access to the private members of the same class they are part of. The improved Fraction class has one each kind.

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

class Fraction
{
  long Numer;    //holds numerator of the fraction
  long Denom;    //holds denominator

  public:

  void Read()
  {
    cin >> Numer >> Denom;
  }

  void Write()
  {
    cout << Numer << '/' << Denom;
  }
};

void main()
{
  Fraction F;

  cout << "Enter the numerator and denominator of a fraction\n";

  F.Read();

  cout << "\nThe fraction is ";

  F.Write();

  cout << endl;
}
```

Note that member functions are accessed by using the **dot operator**, similar to that used when accessing data fields inside a *struct* or using members of *istream*.

Objects and **abstract data** types are implemented in C++ by using classes. The C programming language has structs, which are somewhat different then C++ structs, but C does not have classes. In C++, the only differenct between a struct and a class is that fields of a struct are public by default, while fields of a class are private by default.