

General Instructions:

This program will be your first use of classes and objects. The class named **Fraction** will hold two data members called *top* and *bottom*, both can be of type **int**. The class will have 2 constructors, one the default (no parameters) and the other will take two **int** parameters. You will also implement methods, **display( )**, **add( )**, **mult( )**, and **divide( )**. **ALSO ADD** a single int constructor which creates a Fraction with an assumed denominator = 1. **ALSO ADD** an overloaded operator **+ ( )** which works the same as the **add ( )** method.

Specific Instructions:

The entire program will be a two-module set, nothing new there! The main will be in its own module (**fractiontest.cpp**) , and I have given it to you on the second page. The other module will be the **Fraction** module, that is, the **fraction.h** (header) and the **fraction.cpp** (implementation). The **fraction.h** file is also given to you. Your job, then is to write the **fraction.cpp** file, create a project, and compile and link everything together. Remember to *#include* the **fraction.h** file in both, the **fractiontest.cpp** and **fraction.cpp** files. That is necessary for the compiler.

The class **Fraction** declaration is given below:

```
// File: fraction.h

#include <iostream.h>

#ifndef _FRACTION
#define _FRACTION

class Fraction {
public:

    Fraction ( );           // The two constructors
    Fraction(int,int);

    void display( );       // the other required methods
    Fraction add(Fraction);
    Fraction mult(Fraction);
    Fraction divide(Fraction);
    bool operator < (const Fraction &) const;

private:
    int top;
    int bottom; };

#endif
```

Here is the fractiontest.cpp program file (screen output shown on right) :

```
// File: fractiontest.cpp

#include <iostream.h>
#include "fraction.h"

int main( ) {
    Fraction f(2,5) , g(1,3);
    Fraction h, m(3);

    f.display( );           → (2/5)
    g.display( );           → (1/3)
    h.display( );           → (0/1)

    h = f.mult(g);
    h.display( );           → (2/15)

    h = f.divide(g);
    h.display( );           → (6/5)

    h = f.add(g);
    h.display( );           → (11/15)

    h = m + g;
    h.display( )           → (10/3)

    cout << "Program Over\n";
    return 0; }
```

Required Programming Constructs:

- Class header file, you need to add a method and an operator to the header file
- Class Implementation file
- 2-module project (must be UNIX)
- You must not modify my program
- Documentation (comments for all files), including pre and post conditions for all methods

How Assignment Should be Handed In:

- Code Files (Each of the two modules *separately*, add comment // File: xxxxxx to each file)
- The Output file