Current Assignments

- Project 4 due next Saturday before midnight (Aug 2nd).

  Dinosaur classes.

- Comprehensive Final Exam on Thursday (July 31st)

Final Exam

- Everything that was on the first two exams plus basic classes and objects:
  - Class syntax
  - Private vs. Public
  - Constructors
  - Destructors
  - Class Methods
  - Instantiation of classes
  - Using objects

Sample Problem (find syntax errors)

In Test.h

```
class Test
{
   public
   float Test();
   float Test( float i );
   method1( float x );
   method2( float y);
      ~Test();
        ~Test( int x )
   private
   float var;
}
```

In Test.cpp

```
Test:Test()
{
   var = 0.0;
}
float Test:Test(float i)
{
   var = i;
}
float Test:method1( float x,
   float y )
{
   return x*y*var;
}
float Test:method2( float y )
{
   return (y+var)*y;
}
Test:~Test()
{
}
Test:~Test( int x )
{
```

Sample Problem (find the logic error)

In Test.h

```
class Test
{
   public:
    float method1( float
x );
    float method2( float
y);
   ~Test();

   private:
   Test();
   Test( float i );
   float var;
}
```

In Test.cpp

```
#include "Test.h"
Test::Test()
{
   var = 0.0;
}
Test::Test(float i)
{
   var = i;
}
float Test::method1( float
x )
{
   return x*var;
}
float Test::method2( float
y )
{
   return (y+var)*y;
}
Test::~Test()
{
}
```

Sample Problem (give the output)

In Test.h

```
class Test
{
                public:

Test( float i );
                        float
method1( float x );
                ~Test();

        private:
                float var;
};
```

In Test.cpp

```
#include "Test.h"
Test::Test(float i)
{
   var = i;
}
float Test::method1( float
x )
{
   return x*var;
}
Test::~Test()
{
}
```

In driver.cpp

```
…
#include "Test.h"
int main()
{
   Test a_test( 4.0 );
   cout << "Result: " << a_test.method1( 2.0 ) << endl;
```

Sample Problem

You will be required to write programs like those on the first two exams (including one recursive function).

Example class program:

Write a *Circle* class to represent circles.

Your circle class must store the radius of the circle in a *private* data member called *radius* of type float.

Your circle class must define two *public* methods called diameter ($2r$) and area ($3.14r^2$) that take no arguments and return the diameter and area of the circle defined by radius in your circle object.

Give your circle objects a value for radius when they are created (i.e. with a constructor).

Write a driver program to create two circles of radius 1.0 and 2.0. and which prints the area and diameter of your circles.

Sample Problem (give the output)

In Circle.h

```
class Circle
{
                public:

Circle( int r );
       float diameter();
                        float
area();

       private:
                        float
radius;
};
```

In Circle.cpp

```
#include "Circle.h"
Circle::Circle(float r)
{
   radius = r;
}
float Circle::diameter()
{
   return 2*radius;
}
float Circle::area()
{
   return 3.14*radius*radius;
}
```

In driver.cpp

```
…
#include "Test.h"
int main()
{
   Circle circle1( 2.0 ), circle2(1.0);
      cout  <<  "Area:  "  <<  circle1.area()  <<" "<<
circle1.diameter() << endl;
```

The End - Programming

- Computers and the software they run are becoming ubiquitous.
- In this course you learned how computers are used to solve problems.
- You learned how to use *functions*, *iteration*, *recursion*, and *objects* to solve various problems.
- You wrote software to solve *mathematical* problems, play *games*, store information (a *database*), and *simulate* extinct animals.

The End – C++

- You understand how to write and compile fairly complex programs in C++.
- You have been exposed to all the major elements of C++, variables, control structures, functions, classes, input and output.
- Make sure you put C++ on your resume!