
friend Functions and the this Pointer

3/4/05 CS250 Introduction to Computer Science II 1

Friend Functions

- Only the member functions of a class have direct access to the private data members of the class
- friend functions are friends of the class that are defined outside of the class but still have access to private data members

3/4/05 CS250 Introduction to Computer Science II 2

friend Functions

- The function prototype is placed in the class, preceded by the keyword `friend`
- The function definition can be written anywhere without the class name (`class::`)
- The function is able to directly access the private data members

3/4/05 CS250 Introduction to Computer Science II 3

Class Storage

- functions - only one copy of each function exists in memory independent of the number of objects instantiated using the class declaration
- data members - each unique object of a particular class has space allocated for the data members of the class
- `this` - is a pointer that can be used to access an objects data members. No space associated with the class is allocated for the pointer `this`

3/4/05 CS250 Introduction to Computer Science II 4

Example of this

```
Rational.h
#ifndef RATIONAL_H
#define RATIONAL_H

#include
using namespace std;

class Rational
{
public:
    Rational (int, int);
    print ();

private:
    int numerator;
    int denominator;
};

#endif
```

3/4/05 CS250 Introduction to Computer Science II 5

Example

```
Rational.cpp
#include "Rational.h"

Rational::Rational (int numerator, int denominator)
{
    this->numerator = numerator;
    this->denominator = denominator;
}

Rational::print ()
{
    cout << numerator << '/' << denominator;
}
```

3/4/05 CS250 Introduction to Computer Science II 6

this Pointer

- The **this** pointer is defined as **Rational *** **const** for a nonconstant member function of a class and it's defined as **const Rational *** **const** for a constant member function of a class
- The **this** pointer can be used in cascaded function calls which is where multiple function calls happen in a single statement

3/4/05

CS250 Introduction to Computer Science II

7

Examples

- Let's examine fig. 7.13
- For cascading functions we'll look at fig. 7.14

3/4/05

CS250 Introduction to Computer Science II

8

Summary

- Today we covered
 - Friend Functions
 - this Pointer
- Completed pages 485 - 495

3/4/05

CS250 Introduction to Computer Science II

9