



CPSC 131 – Data Structures

Review and Level Setting

Professor T. L. Bettens

Fall 2020

Agenda

- Pointers
- Composition
- Operator overloading
- Function templates, template functions
- Class templates, template classes
- Shallow vs deep copy

- Source code walkthrough

C++ Pointers

- 1.1 Pointer basics
- 1.2 Operators: new, delete, and ->
- 1.3 Memory regions: Heap/Stack
- 1.4 Memory leaks
- 1.5 Copy constructors
- 1.6 Copy assignment operator
- 1.7 Constructor initializer lists
- 1.8 Rule of three
- 1.9 Destructors
- 1.10 Class templates
- 1.11 Function templates
- 1.12 Range-based for loop
- 1.13 find() function
- 1.14 Short circuit evaluation
- 1.15 Operator overloading
- 1.16 Overloading comparison operators
- 1.17 Overloading stream operators

Classes, Composition, Operator Overloading

- Source code walkthrough

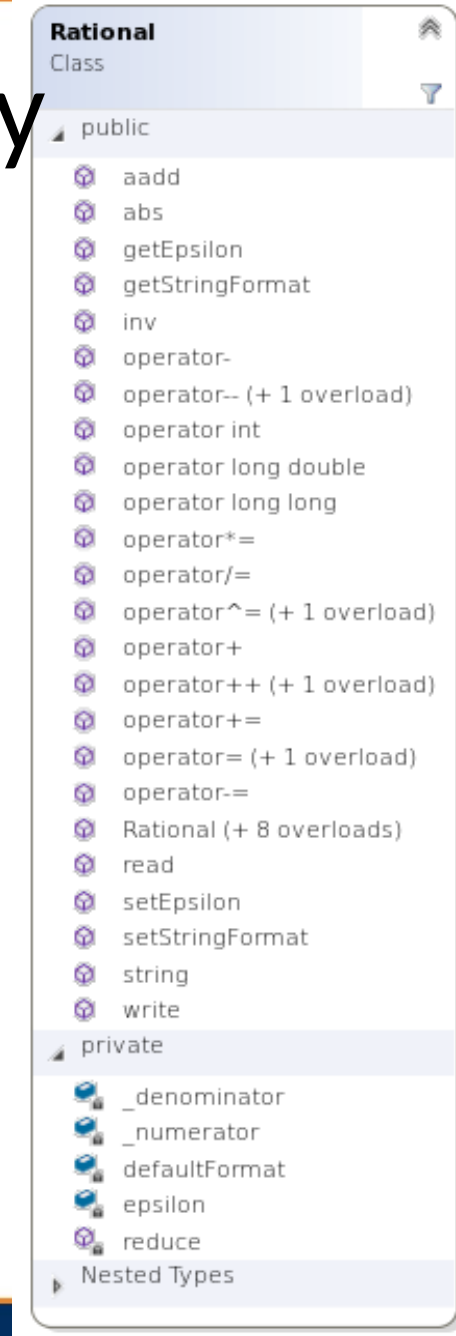
Class Rational and RationalArray Case Study

- Class Rational is a class with value semantics
 - Values are represented as a fraction
 - All arithmetic operations are supported

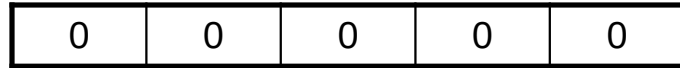
 131 Case Study Review.zip

Source files
Visual Studio Solution Space
VSCode Solution Space

- Class RationalArray is a collection of Rational objects
 - Unbounded array semantics



RationalArray container class



Contains pointer-to-Rational, initialized to 0 (null). Guessed there would be 5 elements. Your constructor would do something like:

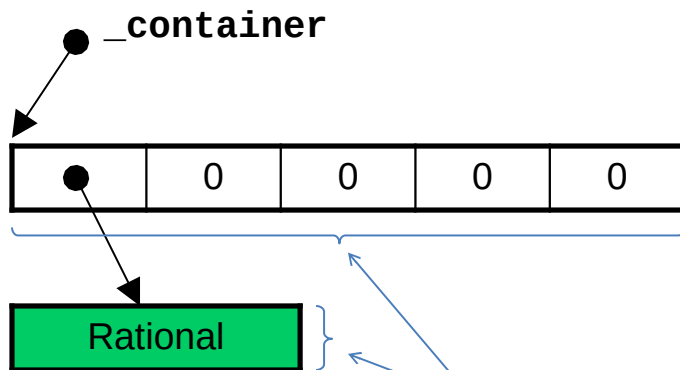
`_array = new Rational [5];`*

Usage:

```
RationalArray myArray; // _size = 0, _capacity = 5
```

Now, let's add an element:

```
Rational number1;  
myArray.append(number1)
```



append() needs to:

- 1. Allocate new storage for the additional object*
- 2. Copy the contents of number1 to the newly allocated Rational object*
- 3. Link the new object into the collection*
- 4. The end result looks like the left*
- 5. The size of the container is now 1*

both dynamically allocated and maintained on the heap

Concept:

```
append( Rational const & r )  
{  
    if(_size < _capacity) _container[_size++] = new Rational(r); // uses copy constructor  
    else { ... }  
}
```

RationalArray
Class

public

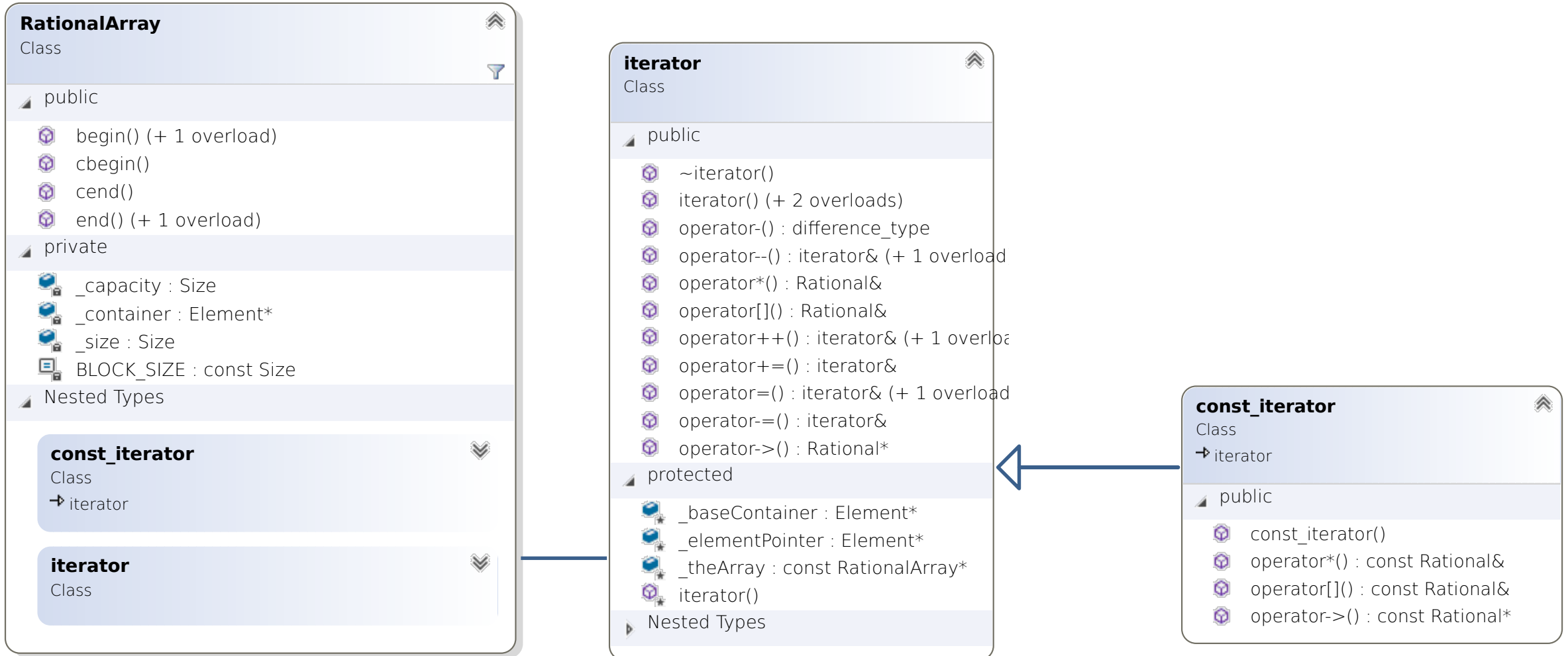
- append() : Index (+ 1 overload)
- insert() : Index (+ 1 overload)
- operator[]() : Rational& (+ 1 overload)
- prepend() : Index (+ 1 overload)
- RationalArray() (+ 7 overloads)
- remove() : Index
- replace() : RationalArray& (+ 1 overload)

private

- _capacity : Size
- _container : Element*
- _size : Size

Nested Types

RationalArray Iterator Support



Source code walkthrough (1)

Topic	Reference main.cpp
Anonymous namespaces	49
Templated aliases	52
Random numbers	72,80,84
Classes	61-85
• Access	63,75
• Instance attributes	83,84
• Class attributes	80
• Enumerations	77

Topic	Reference main.cpp
Constructor initialization lists	65-66
Tuples	52, 64, 106
Range based for loops	111,125
Exception catching	104,189
Integer overflow	All
This driver:	
• creates an array of random fractions,	111
• Creates a reverse copy of the array in a vector	115
• Uses rational and floating point arithmetic to total the elements in both directions – left to right, and right to left.	125-135
• Observe floating point accumulated roundoff errors vs exact rational arithmetic, and integer overflow as fractions are normalized	

Source code walkthrough (2)

Topic	Reference Library/Rational.hpp
Include guard	26, 27,410
Named namespaces	39
Friends	48-51
Operator overloading <ul style="list-style-type: none">• Insertion and extraction operators• Relational operators• Unary operators• Binary operators• Casting operators	50,51 48,361-377 145-165 202-224 119-122
Exception hierarchy	62-67

Topic	Reference Library/Rational.hpp
Static member functions	244-248
Constructors <ul style="list-style-type: none">• Conversion constructors• Copy constructor and copy assignment• Move constructor and move assignment	85-92 105,106 109,110
User defined literals	384-397
User defined manipulators with argument	260-272

Source code walkthrough (3)

Topic	Reference Library/Rational.hxx
Out-of-class inline function definitions	all
Delegating constructors	58, 59, 65, 69
Overloaded operator implementation	all

Topic	Reference Library/Rational.cpp
Function templated	34, 64, 89
In-function static objects	91
Static member functions	108-110
Throwing exceptions	129 171,...

Source code walkthrough (4)

Topic	Reference Library/RationalArray.hpp
Templated constructors	94,99
User defined iterators	120-130
Member attribute initialization	214-216
Hide vs override virtual functions	337-348
Pointer arithmetic	351-356
Dynamically allocated array of elements	216

Topic	Reference Library/RationalArray.hxx
Out-of-class function template definitions	68, 71, ...

Source code walkthrough (5)

Topic	Reference Library/RationalArray.cpp
Dynamic array of pointers to dynamically allocated objects	252-294
Deep copy vs shallow copy	76-92, 179
Clear a dynamically allocated container	130
Updating an object through a pointer	151

Topic	Reference Library/RationalArray.cpp
Matching new/delete & new[]/delete[]	265/272
Pointer, pointer-to-object, pointer-to pointer-to-object	252-294
Allocating arrays of value-initialized objects	272