CS100
Introduction to Programming

Tutorial 9: shared pointers, friend, factory method

Part 0
Debugging

Understanding Errors

```
hw2.c:87:7: error: ‘foo’ undeclared
```

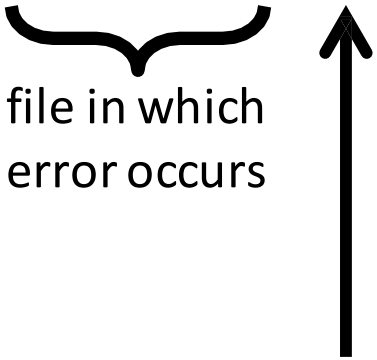
Understanding Errors

```
hw2.c:87:7: error: ‘foo’ undeclared
```

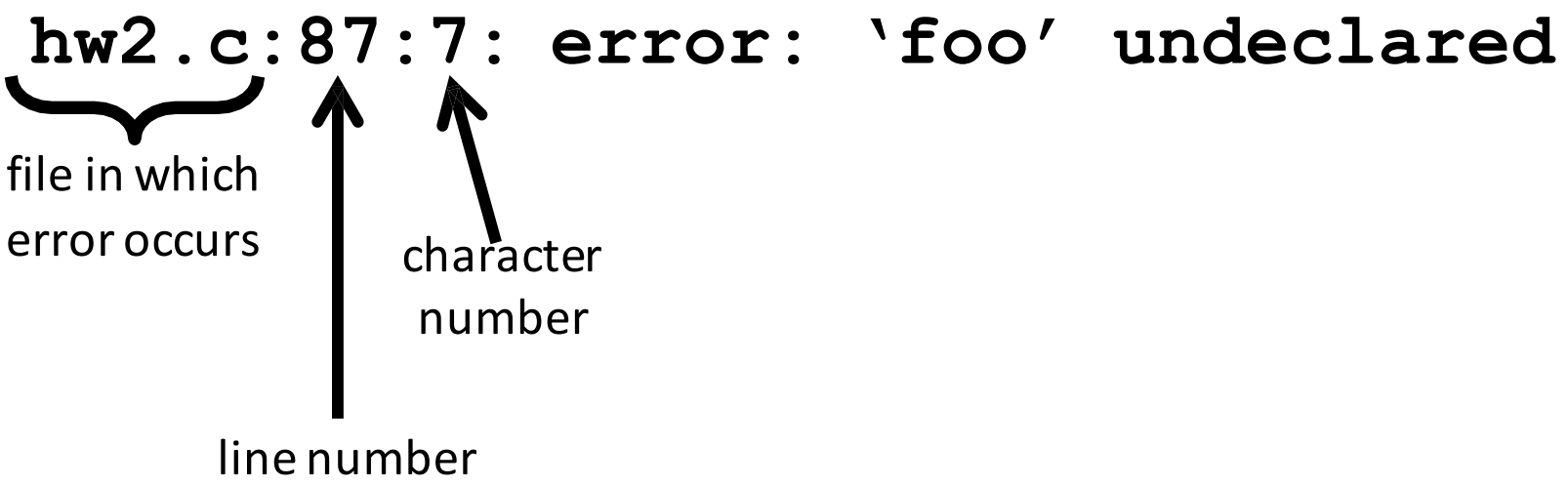
file in which
error occurs

Understanding Errors

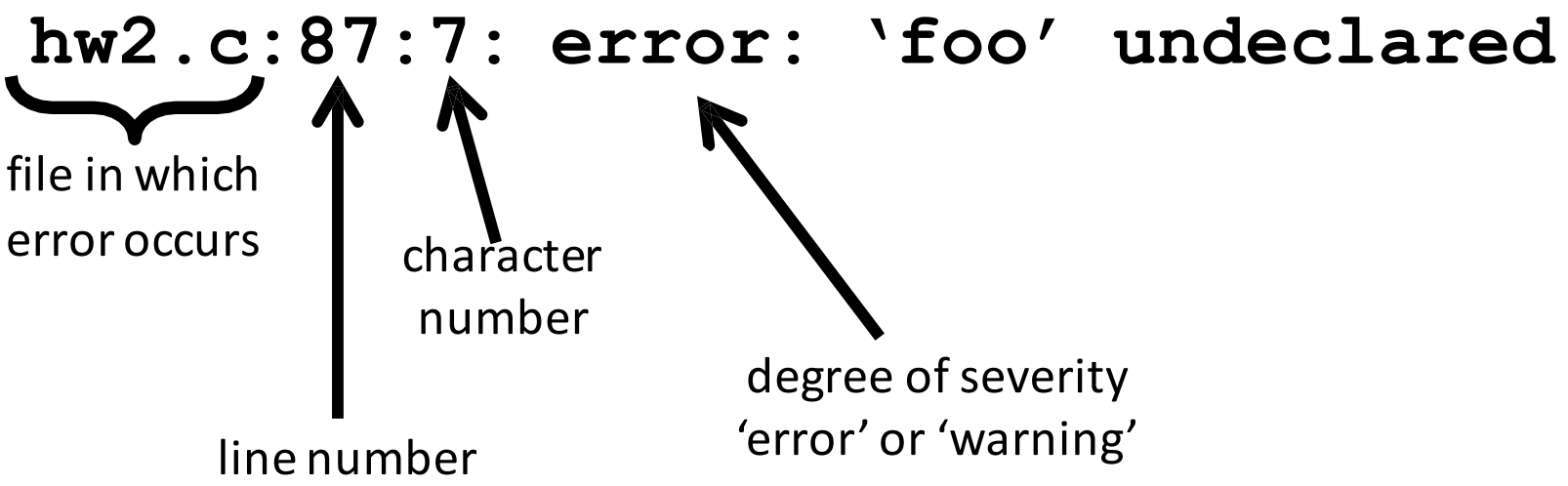
```
hw2.c:87:7: error: ‘foo’ undeclared
```



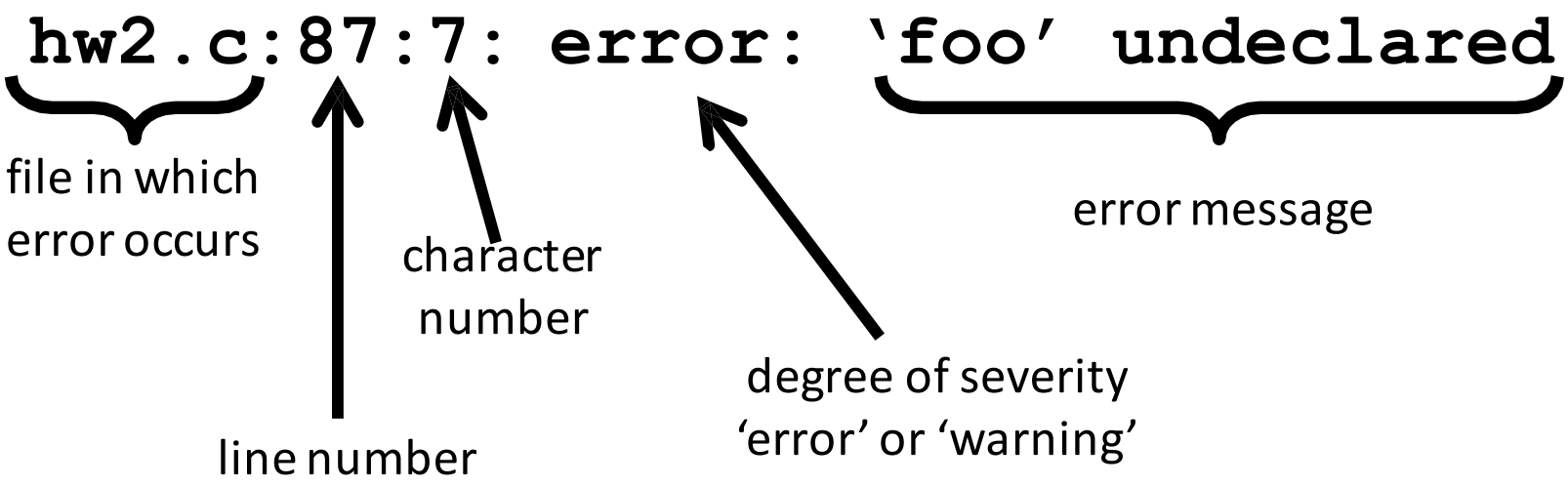
Understanding Errors



Understanding Errors



Understanding Errors



#1 Rule of Debugging

- start with the **very first** error or warning

- recompile every time an error is fixed
  - errors will cascade
  - and de-cascade when fixed!

Cascading Errors

```
int numStudnts;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

Cascading Errors

```
int numStudnts;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

```
> gcc -Wall average.c
```

Cascading Errors

```
int numStudnts;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

```
> gcc -Wall average.c
```

- the **`-Wall`** flag shows all of warnings

Cascading Errors

```
int numStudnts;
for (i = 0; i < numStudents; i++) {
  total += grades[i];
}
avg = total/numStudents;
```

```
> gcc -Wall average.c
average.c:5:5: warning: unused variable ‘numStudnts’
average.c:22:17: error: ‘numStudents’ undeclared
average.c:25:13: error: ‘numStudents’ undeclared
```

Cascading Errors

```
int numStudnts;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

```
> gcc -Wall average.c
average.c:5:5: warning: unused variable ‘numStudnts’
average.c:22:17: error: ‘numStudents’ undeclared
average.c:25:13: error: ‘numStudents’ undeclared
```

Cascading Errors

```
int numStudnts;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

```
> gcc -Wall average.c
average.c:5:5: warning: unused variable 'numStudnts'
average.c:22:17: error: 'numStudents' undeclared
average.c:25:13: error: 'numStudents' undeclared
```

Cascading Errors

```
int numStudents;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

Cascading Errors

```
int numStudents;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

```
> gcc -Wall average.c
```

Cascading Errors

```
int numStudents;
for (i = 0; i < numStudents; i++) {
   total += grades[i];
}
avg = total/numStudents;
```

```
> gcc -Wall average.c
```

- got rid of all 3 errors!

When Errors Occur

- compile time
  - pretty easy (normally typos or simple mistakes)
- linking
  - slightly harder (could be easy, could require rethinking how your code is laid out)
- run time
  - often difficult to pinpoint, and sometimes hard to spot at all
  - best bet is to use a debugger

Common Compiler Errors

`hw2.c:87:7: error: ‘foo’ undeclared`

- if **`foo`** is a **variable**:
  - forgot to declare
  - misspelled (on declaration or on use)
- if **`foo`** is a **function**:
  - forgot to **`#include`** file containing the prototype
  - misspelled (on declaration or on use)

Common Compiler Errors

```
hw2.c:37:6: warning: unused variable
    'bar'
```

- variable was declared but not used
  - normally because variable declaration has a typo
  - if you're in the midst of writing code, this warning may be *temporarily* acceptable
    - haven't had a chance to use the variable yet

Common Compiler Errors

```
hw2.c:54: warning: suggest
    parentheses around assignment
    used as truth value
```

- often a mistake inside a control statement
  - you meant to use **==** not **=**
  - (you want equivalency, not assignment)

Common Compiler Errors

```
hw2.c: 51: error: expected ';'
    before 'for'
```

- missing semicolon on previous line of code
- 'for' is simply the word directly following the missing semicolon
  - could be 'int' or 'if' or a variable name, etc

Common Linker Errors

```
hw4.o: In function ‘main’:
hw4.c:91: undefined reference to ‘Fxn’
```

- linker can’t find code for ‘Fxn’ in any .o file
  - forgot to link `.o` file
  - misspelled named of Fxn
  - parameter list is different
    - differences between prototype/definition/call

Common Linker Errors

```
/usr/lib64/gcc/[...]/crt1.o: In function
    `_start':
/home/[...]/start.S:119: undefined
    reference to main
```

- you compiled a file that does not contain a **`main()`**
- without using the **`-c`** flag to indicate separate compilation

Error messages can be very long …

```
> gcc -Wall structs.c
In file included from /usr/include/stdio.h:33:0,
                 from structs.c:6:
/usr/lib64/gcc/x86_64-suse-linux/4.7/include/stddef.h:213:1: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'typedef'
In file included from /usr/include/stdio.h:74:0,
                 from structs.c:6:
/usr/include/libio.h:307:3: error: unknown type name 'size_t'
/usr/include/libio.h:311:67: error: 'size_t' undeclared here (not in a function)
/usr/include/libio.h:339:62: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/libio.h:348:6: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/libio.h:470:19: error: expected '=', ',', ';', 'asm' or '__attribute__' before '_IO_sgetn'
In file included from structs.c:6:0:
/usr/include/stdio.h:319:35: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:325:47: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:337:20: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:344:10: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:386:44: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:390:45: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:666:11: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:669:9: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:679:8: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdio.h:709:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'fread'
/usr/include/stdio.h:715:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'fwrite'
/usr/include/stdio.h:737:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'fread_unlocked'
/usr/include/stdio.h:739:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'fwrite_unlocked'
In file included from structs.c:9:0:
/usr/include/string.h:43:8: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:46:56: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:55:18: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:62:42: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:65:56: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:92:48: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:129:39: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:137:9: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:143:57: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:150:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'strxfrm'
In file included from structs.c:9:0:
/usr/include/string.h:165:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'strxfrm_l'
/usr/include/string.h:180:45: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:281:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'strcspn'
/usr/include/string.h:285:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'strspn'
/usr/include/string.h:395:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'strlen'
/usr/include/string.h:402:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'strnlen'
/usr/include/string.h:423:12: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:447:33: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:451:53: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:455:31: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:458:54: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:536:61: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:573:34: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/string.h:576:39: error: expected declaration specifiers or '...' before 'size_t'
In file included from structs.c:11:0:
/usr/include/stdlib.h:139:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before '__ctype_get_mb_cur_max'
In file included from structs.c:11:0:
/usr/include/stdlib.h:331:4: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:361:4: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:465:22: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:467:22: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:467:38: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:479:36: error: expected declaration specifiers or '...' before 'size_t'
In file included from /usr/include/stdlib.h:491:0,
                 from structs.c:11:
/usr/include/alloca.h:32:22: error: expected declaration specifiers or '...' before 'size_t'
In file included from structs.c:11:0:
/usr/include/stdlib.h:497:22: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:502:45: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:502:65: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:755:9: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:755:25: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:760:34: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:760:50: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:839:6: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:842:6: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:846:31: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:850:31: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:859:36: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:863:34: error: expected declaration specifiers or '...' before 'size_t'
/usr/include/stdlib.h:870:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'mbstowcs'
/usr/include/stdlib.h:873:15: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'wcstombs'
```

… but not too hard to fix

- Follow the message til the original calling point
  - …
  - In file included from …
  - …
  - In file included from …
  - …
  - Instantiated here …
  - …
  - Instantiated here …
  - Error message

Debugging Basics

- if the error's not clear from just looking at the code, you can try:

- inserting probe statements with printf
  - (but adding a printf might change your error!)
- rubber duck debugging
- googling the error message
- using a debugger

Debuggers

- see what is going on “inside” the program
  - more powerful and accurate than printf() probes
- examine individual variables (value & address)
  - can change variable’s value on the fly
- step through code line by line
  - can skip blocks of code you don’t want to see

Using GDB

- must use the '**`-g`**' flag when compiling

- open program for testing using command line:
  `gdb hw2`

- GDB – Gnu Project Debugger (text based)

Using GDB

- debugger allows you to:
  - add breakpoints to stop the program at specific points
  - use ‘print’ or ‘display’ to show values (or addresses) of variables
  - step through code line by line

Part 1
shared pointers

Shared pointers

- Imagine the following simple Test class
  - The class does not do much except notifying us about construction and destruction

```
class TestClass {
public:
  TestClass() {
    std::cout << "dummy object is created\n"; }
  virtual ~TestClass() {
    std::cout << "dummy object is destroyed\n"; }
  void printSomething() { std::cout << "hello\n"; }
private:
  double m_dummy;
};
```

Shared pointers

- The class is used in the main function

```
int main() {
  int indexA = 1;
  int indexB = 1;
  if( indexA == 1) {
    std::cout << "Entering scope A\n";
    SharedPointer<TestClass> ptrA;
    if( indexB == 1 ) {
      std::cout << "Entering scope B\n";
      SharedPointer<TestClass> ptrB( new TestClass() );
      ptrA = ptrB;
    }
    std::cout << "Returning to scope A\n";
  }
  std::cout << "Returning to global scope\n";
  return 0;
}
```

Custom shared-pointer class

Shared pointers

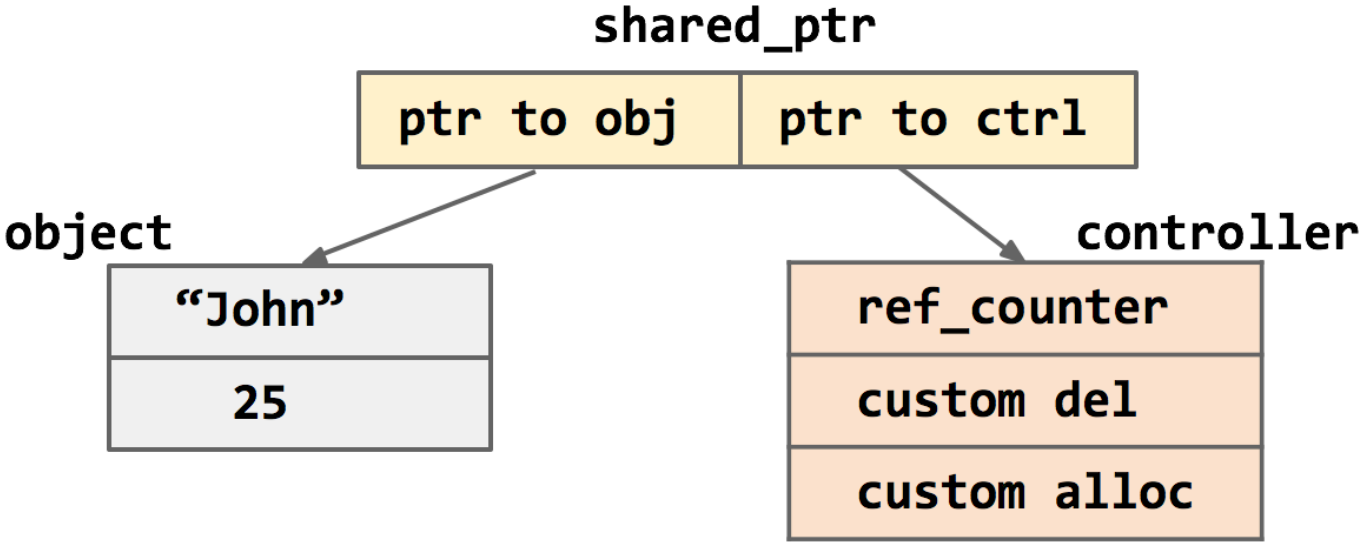
- The shared pointer has the declaration

```
template<class T>
class SharedPointer {
public:
  SharedPointer();
  SharedPointer( T * ptr );
  SharedPointer( const SharedPointer & ptr );
  virtual ~SharedPointer();

  const SharedPointer & operator=( const SharedPointer & ptr );

  T * get();
  T & operator*();
  T * operator->();
private:
  T * m_ptr;
  SharedPointerController * m_controller;
};
```

Remember: A shared pointer takes two members, a pointer to the underlying variable, and a pointer to a control structure with reference counter

std::shared_ptr

- Shared pointers have a garbage collection mechanism based on a reference counter contained in a control block
- Each new shared owner copies the pointer to the control block and increases the count by 1



Shared pointers

- Tasks:
  - Implement the Shared pointer
  - Implement the control block

Part 2

friends

Using Friend

- We have the following very simple vector class (Essentially just a wrapper around std::vector)

```
class MyVector {
public:
  MyVector();
  virtual ~MyVector();

  double & at( int index );
  const double & at( int index ) const;
  void clear();
  int size() const;
  MyVector & push_back( double val );
private:
  std::vector<double> m_data;
};
```

Using Friend

- Task
  - Make it possible to stream a MyVector to an std::ostream (base class of std::ofstream, std::cout, std::stringstream, etc.)
  - Make it possible for this function to have direct access to MyVector's data (by using friend)

Part 3

The factory method

Factory method

- Let Number be the base class of a factory that can produce random numbers of different types

```
class Number {
public:
  static Number * make_number(int type);

  Number( int type ) : m_type(type) {};

  virtual Number * plus( Number * nbr ) = 0;
  virtual void print() = 0;
  int type() { return m_type; };
private:
  int m_type;
};
```

Factory method

- Number has a few particularities:
  - The factory is implemented as a static method inside Number
  - Whenever a derivative of Number is created, the type (called choice in the lecture) is passed to the constructor and stored in a member variable
  - Number has three methods
    - A concrete one which returns type
    - A purely virtual one to print the number to the console
    - A purely virtual one to add this plus another Number and return a new Number which represents the sum

Factory method

- Task:
  - Make it possible that the factory can produce one of two different kinds of objects
    - RandomRealNumber (if choice/type == 1)
    - RandomRationalNumber (if choice/type != 1)
  - Implement these derivates of Number
  - Make sure that they are safe
    - If two Numbers are added, make sure that they are compatible!

Factory method

- Example test code:

```
int main() {
  Number * nbr1 = Number::make_number(1);
  Number * nbr2 = Number::make_number(1);
  Number * nbr3 = Number::make_number(2);
  Number * nbr4 = Number::make_number(2);
  nbr1->print();
  nbr2->print();
  nbr3->print();
  nbr4->print();

  Number * res1 = nbr1->plus(nbr2);
  Number * res2 = nbr3->plus(nbr4);
  res1->print();
  res2->print();
  return 0;
}
```