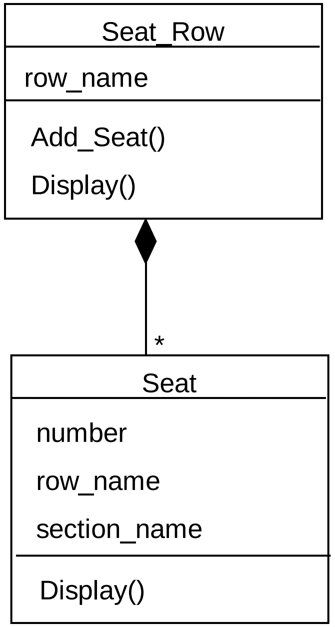
Looking Up

How an object can have a link to the object in which it is contained

Composition

- We have several cases of an object that has a collection of subordinate objects.



"Composition"

or "Has a"

Relationship

Seat_Row.h

- As implemented in C++, the Seat_Row object has an array of pointers to Seat.

```
#include "Seat.h"
 ...
Seat* seats[MAX_SEATS_PER_ROW];
```

- It can aways get to any of its contained Seat objects.

Seat.h

- But a Seat object cannot get to the Seat_Row object that contains it.

- Has row name as a member.
  - Bad!

- This information should only be known in one place.
  - The Seat_Row object

- If the Seat object could access its Row it would not need a copy of the row name.

Upward Navigation

- So far we have not been able to do this.

- The obvious solution does not work.
  - Let Seat.h #include Seat_Row.h.
  - Provide a member variable that points to the Seat's containing Seat_Row.

Try it!

- Download the solution for Project 3

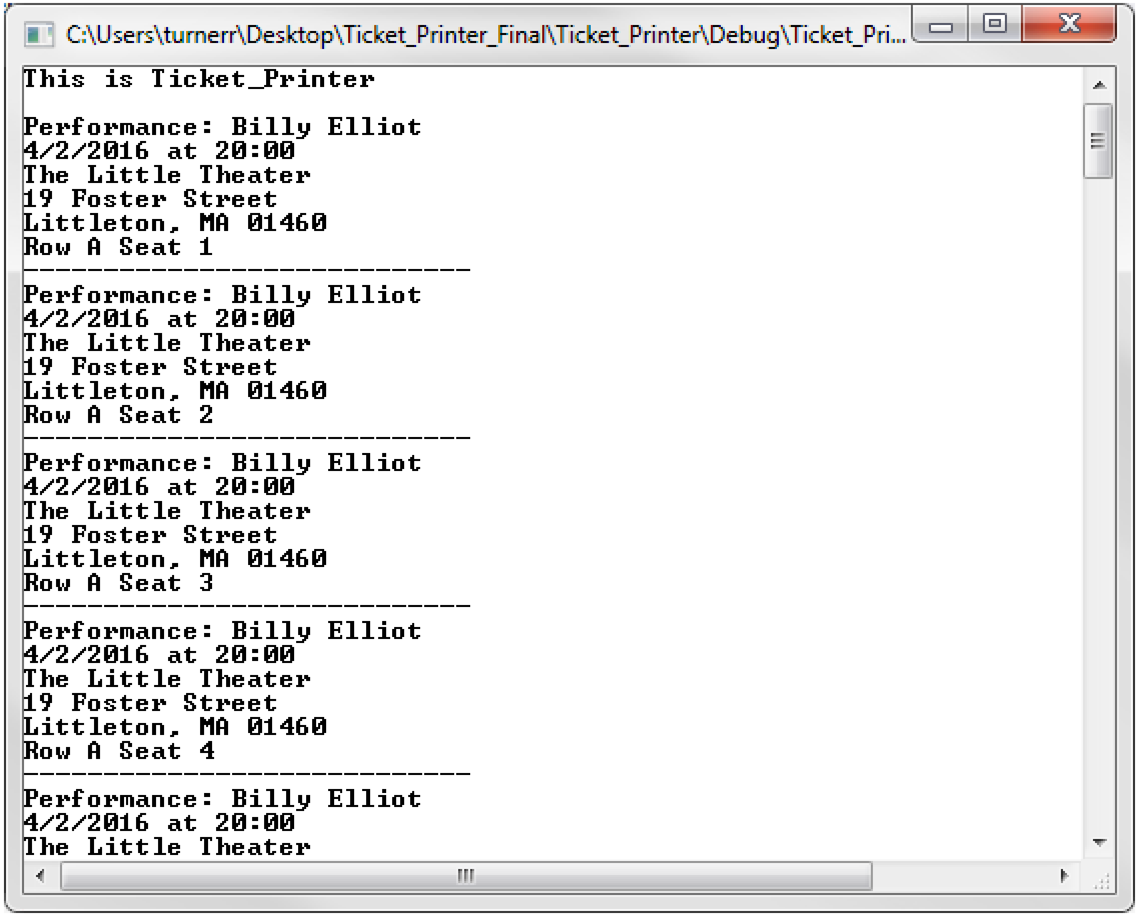
  http://www.cse.usf.edu/~turnerr/Object_Oriented_Design/Downloads/2016_02_19_In_Class/

  File Ticket_Printer_Final.zip

- Expand
  - Rename folder Upward_Navigation

- Build and run

Ticket_Printer



C:\Users\turnerr\Desktop\Ticket_Printer_Final\Ticket_Printer\Debug\Ticket_Pri...

```
This is Ticket_Printer

Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Row A Seat 1
----------------------------
Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Row A Seat 2
----------------------------
Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Row A Seat 3
----------------------------
Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Row A Seat 4
----------------------------
Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
```

Seat.h

- In Seat.h add #include “Seat_Row.h”

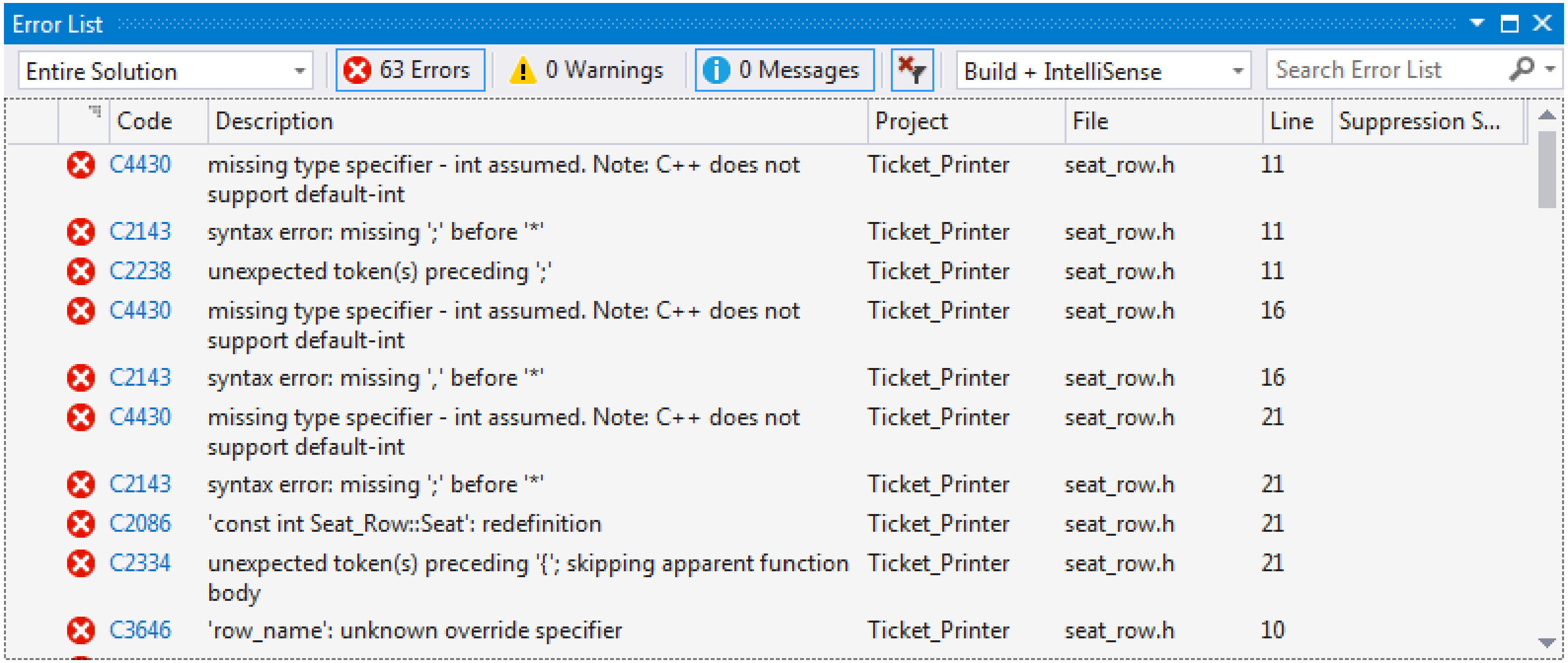
```
#pragma once
#include <string>
#include "Seat_Row.h"

using std::string;

class Seat
{
private:
    Seat_Row* seat_row;
    ...
}
```

- Rebuild

Seat.h



	Code	Description	Project	File	Line	Suppression S...
✖	C4430	missing type specifier - int assumed. Note: C++ does not support default-int	Ticket_Printer	seat_row.h	11	
✖	C2143	syntax error: missing ';' before '*'	Ticket_Printer	seat_row.h	11	
✖	C2238	unexpected token(s) preceding ';'	Ticket_Printer	seat_row.h	11	
✖	C4430	missing type specifier - int assumed. Note: C++ does not support default-int	Ticket_Printer	seat_row.h	16	
✖	C2143	syntax error: missing ',' before '*'	Ticket_Printer	seat_row.h	16	
✖	C4430	missing type specifier - int assumed. Note: C++ does not support default-int	Ticket_Printer	seat_row.h	21	
✖	C2143	syntax error: missing ';' before '*'	Ticket_Printer	seat_row.h	21	
✖	C2086	'const int Seat_Row::Seat': redefinition	Ticket_Printer	seat_row.h	21	
✖	C2334	unexpected token(s) preceding '{'; skipping apparent function body	Ticket_Printer	seat_row.h	21	
✖	C3646	'row_name': unknown override specifier	Ticket_Printer	seat_row.h	10	

Circular dependencies

Seat.h includes Seat_Row.h

But Seat_Row.h includes Seat.h.

Expects Seat.h to be in front of it, so that Seat is already defined.

Upward Navigation?

- If we can’t include Seat_Row.h in Seat.h how can class Seat have a pointer to the Seat_Row that includes it?

Providing a Seat_Row Pointer

Seat.h

```
#pragma once
#include <string>
using namespace std;

class Seat_Row;
```

An *incomplete* declaration.

```
class Seat
{
private:
    // string seat_row_name
    const Seat_Row* const seat_row;    // Upward Pointer
    int seat_number;
public:
    Seat (const Seat_Row* const Seat_Row_,
          int Seat_Number);

   string Seat_Row_Name() const;

    void Display() const;
};
```

Incomplete Declaration

- Just tells compiler that Seat_Row is a class.

- We cannot instantiate a Seat_Row here.
  - Compiler does not know the class definition.

- We *can* declare a Seat_Row* here.
  - All the compiler needs to know is the fact that Seat_Row is a class.

- We cannot dereference the pointer here.
  - Compiler needs the class defintion

Providing a Seat_Row Pointer

- It should be a const pointer to const.
  - A Seat should not be able to modify its row!

```
const Seat_Row* const seat_row;
```

- We have to initialize it with the constructor’s initializer list.

Seat.cpp

```
#include <iostream>
#include "Seat.h"
#include "Seat_Row.h"

using namespace std;

Seat::Seat(const Seat_Row* const Seat_Row_,
           int Seat_Number) :
           seat_row(Seat_Row_),
           seat_number(Seat_Number)
{}
```

It is OK to include Seat_Row.h in the .cpp file.

This does not create a circular dependency.

Seat.cpp

```
string Seat::Seat_Row_Name() const
{
    return seat_row->Row_Name();
}

void Seat::Display() const
{
    cout << "Seat " << seat_row->Row_Name()
         << seat_number << endl;
}
```

We can dereference the Seat_Row* here.

We now have the class definition.

We have to add Row_Name
method to class Seat_Row

Seat_Row.h

Add at top

```
#include <string>
```

Add at end

```
 const string Row_Name() const { return row_name;}
```

Ripple Effects

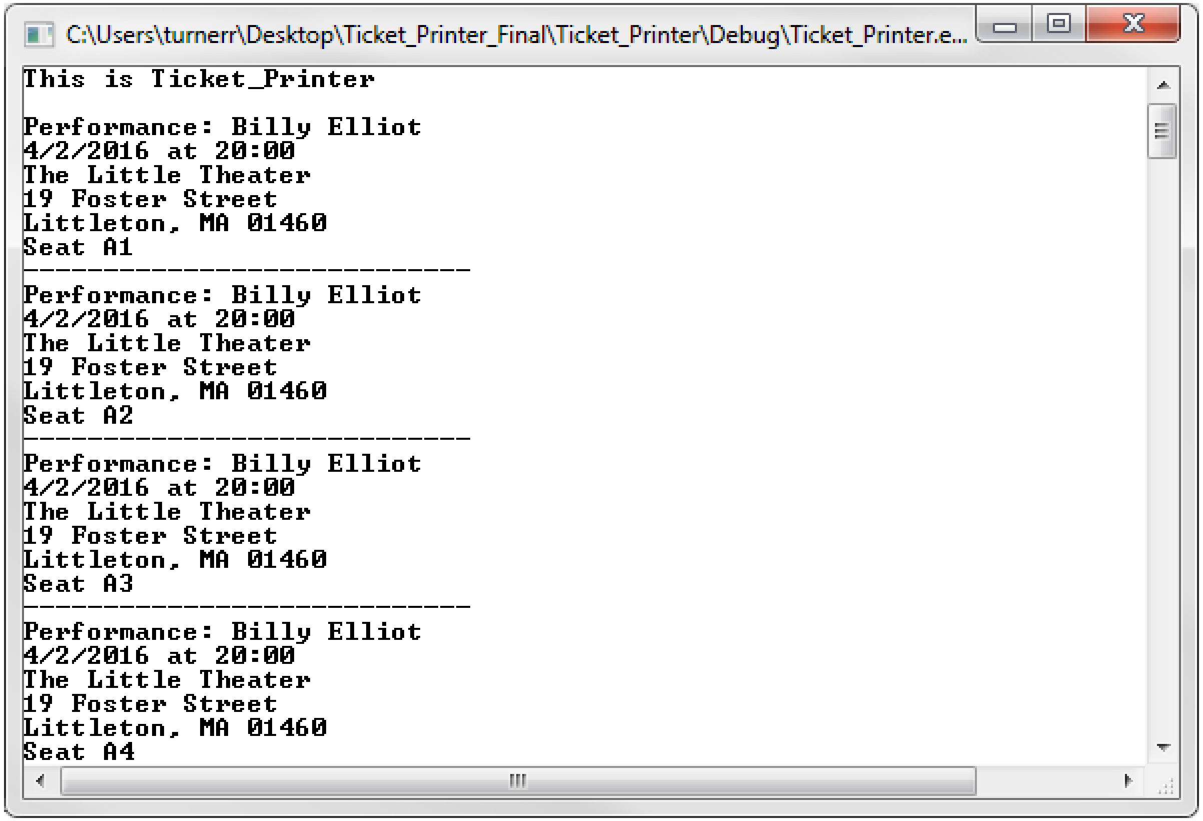
- We have to update function Create_Seat_Row in main.cpp to use the new constructor for Seat.

main.cpp

```
Seat_Row* Create_Seat_Row(const string seat_row_name,
                          int number_of_seats)
{
    Seat_Row* row = new Seat_Row(seat_row_name);
    for (int i = 1; i <= number_of_seats; ++i)
    {
        Seat* new_seat = new Seat(row, i);
        row->Add_Seat(new_seat);
    }
    return row;
}
```

Build and run

Works the Same



C:\Users\turnerr\Desktop\Ticket_Printer_Final\Ticket_Printer\Debug\Ticket_Printer.e...

```
This is Ticket_Printer

Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Seat A1
----------------------------
Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Seat A2
----------------------------
Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Seat A3
----------------------------
Performance: Billy Elliot
4/2/2016 at 20:00
The Little Theater
19 Foster Street
Littleton, MA 01460
Seat A4
```

End of Presentation