



# Show from XML

---



# Objectives

---

- You will be able to
  - Write C++ code to convert the DOM for a Show into a C++ object.



# Show.xml

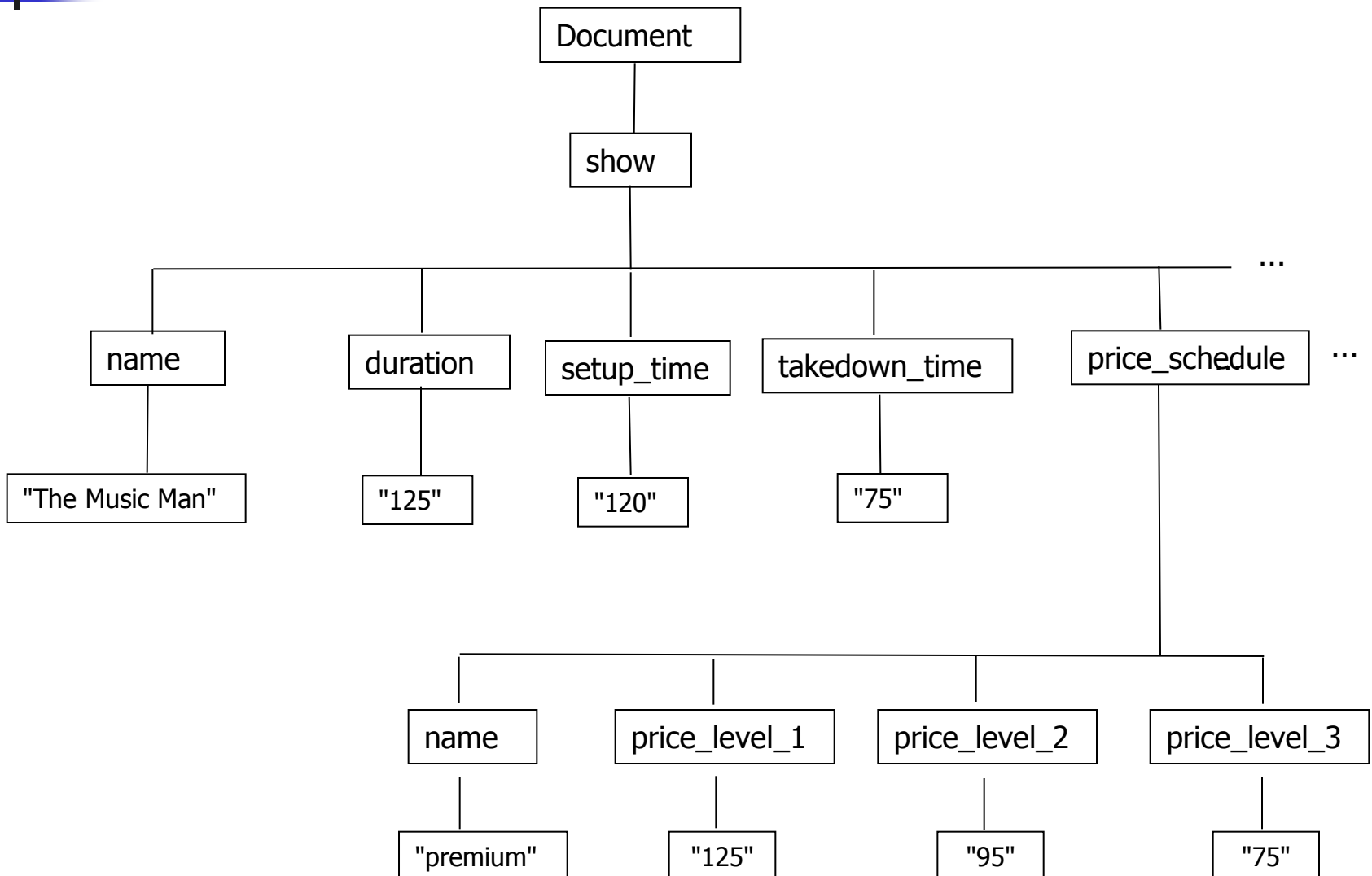
---

- Download if you don't already have it:
- [http://www.csee.usf.edu/~turnerr/Object\\_Oriented\\_Design/Downloads/2016\\_03\\_01\\_XML/](http://www.csee.usf.edu/~turnerr/Object_Oriented_Design/Downloads/2016_03_01_XML/)
  - File Show.xml

# Show.xml

```
Show.xml  [X]
1  <?xml version="1.0" encoding="utf-8" ?>
2  <show>
3      <name>The Music Man</name>
4      <duration>135</duration>
5      <setup_time>120</setup_time>
6      <takedown_time>75</takedown_time>
7
8      <price_schedule>
9          <name>Premium</name>
10         <price_level_1>125</price_level_1>
11         <price_level_2>95</price_level_2>
12         <price_level_3>75</price_level_3>
13     </price_schedule>
14
15     <price_schedule>
16         <name>Normal</name>
17         <price_level_1>95</price_level_1>
18         <price_level_2>80</price_level_2>
19         <price_level_3>65</price_level_3>
20     </price_schedule>
21
22     <price_schedule>
23         <name>Economy</name>
24         <price_level_1>75</price_level_1>
25         <price_level_2>60</price_level_2>
26         <price_level_3>45</price_level_3>
27     </price_schedule>
28 </show>
```

# Show.xml



# New Project

**New Project**

Recent | .NET Framework 4.5 | Sort by: Default | Search Installed Templ

**Installed**

- Templates
  - Visual C#
  - Visual Basic
  - Visual F#
  - Visual C++
    - Windows
      - ATL
      - CLR
      - General
      - MFC

**Win32 Console Application** Visual C++  
A project for creating a Win32 console application

**Win32 Project** Visual C++

[Click here to go online and find templates.](#)

Online

Name: Show\_from\_XML

Location: C:\Users\Rollins\Desktop **Browse...**

Solution: Create new solution


Solution name: Show\_from\_XML

☒ Create directory for solution  
☐ Add to source control

**OK** **Cancel**

# New Project

Win32 Application Wizard - Show\_from\_XML

 **Application Settings**

Overview  
Application Settings

Application type:

- ☐ Windows application
- ☒ Console application
- ☐ DLL
- ☐ Static library

Additional options:

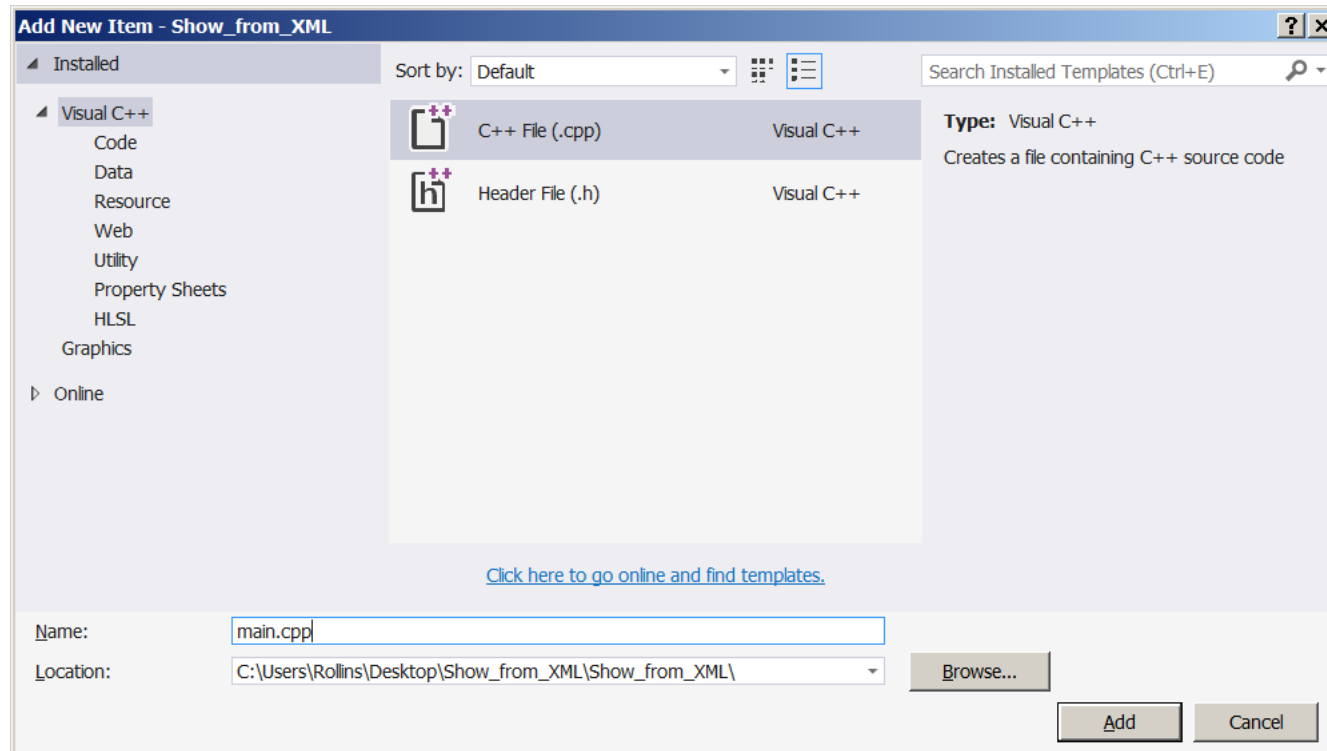
- ☒ Empty project
- ☐ Export symbols
- ☐ Precompiled header
- ☐ Security Development Lifecycle (SDL) checks

Add common header files for:

- ☐ ATL
- ☐ MFC

< Previous    Next >    Finish    Cancel

# Add New Item







# main.cpp

---

```
#include <iostream>

using namespace std;

int main (void)
{

    cout << "This is Show_from_XML\n";

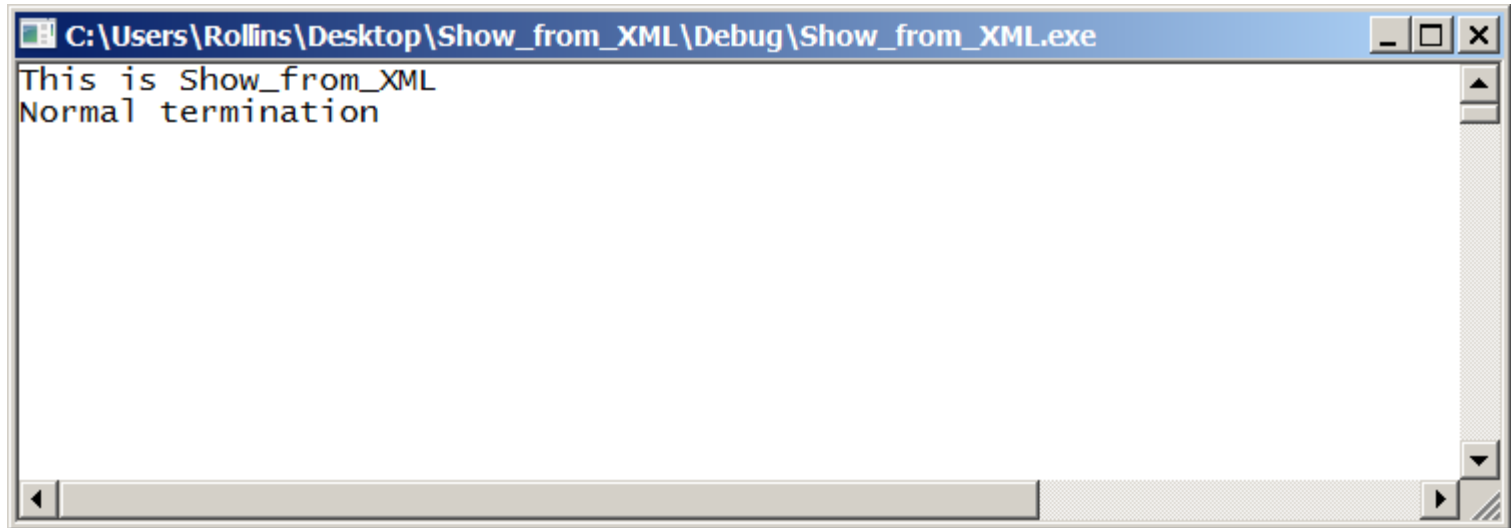
    cout << "Normal termination\n";
    cin.get();
    cin.get();
    return 0;
}
```

## Build and run



# We have a working program!

---



```
C:\Users\Rollins\Desktop\Show_from_XML\Debug\Show_from_XML.exe
This is Show_from_XML
Normal termination
```



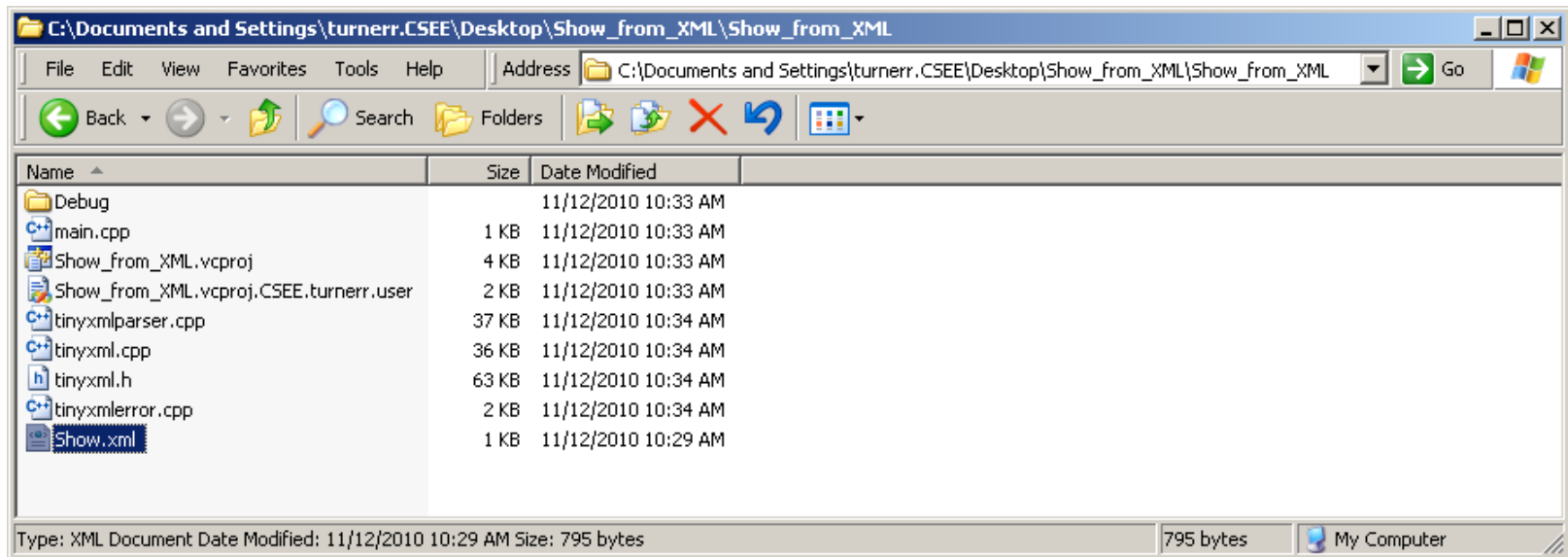
# Get the XML Parser

---

- Download program from class web site:
- [http://www.csee.usf.edu/~turnerr/Object\\_Oriented\\_Design/Downloads/2016\\_03\\_01\\_XML/](http://www.csee.usf.edu/~turnerr/Object_Oriented_Design/Downloads/2016_03_01_XML/)
  - File test\_venue\_xml.zip
- Expand
- Copy tinyxml files into new project directory.
  - tinyxmlparser.cpp
  - tinyxml.cpp, .h
  - tinyxmlerror.cpp
- Add to project.
- Add `#include "tinyxml.h"` to main.cpp

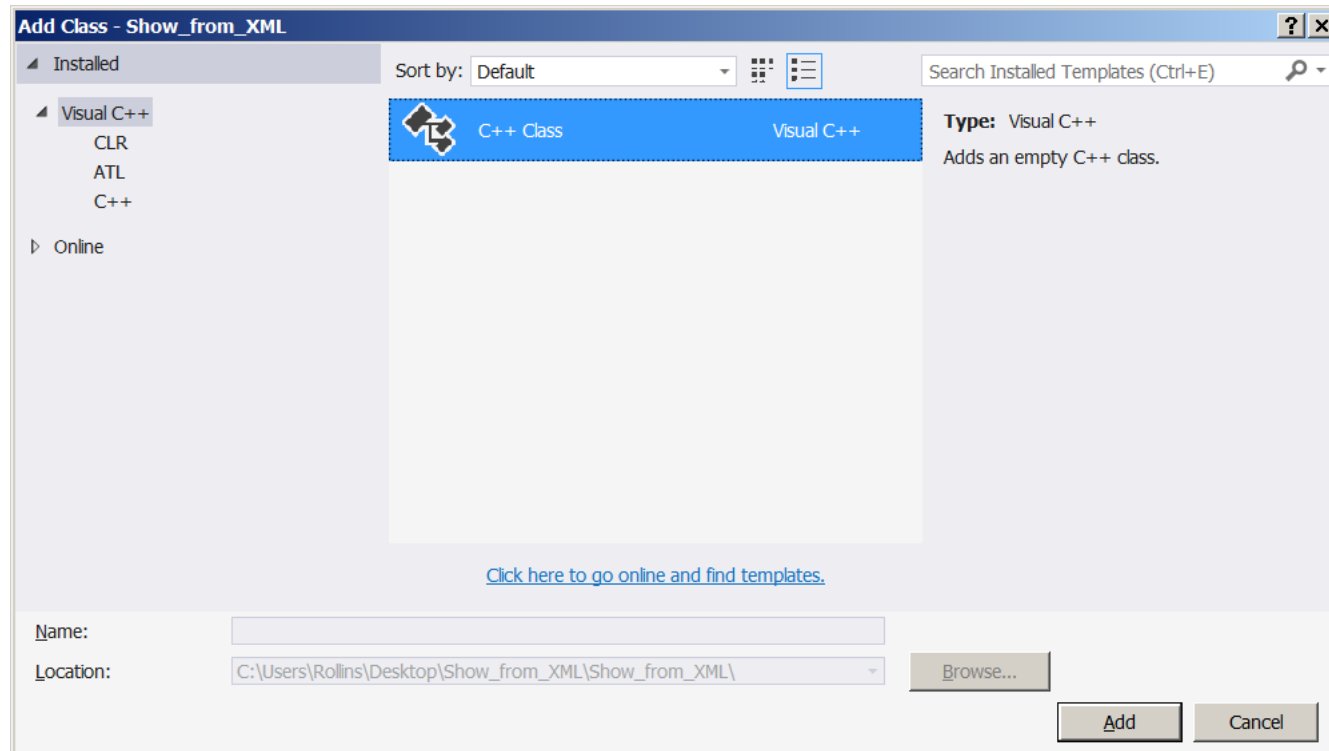
# Show.xml

- Move Show.xml into the project directory.
  - Same folder as the source code.
  - This will be the default directory when we run the program under Visual Studio.

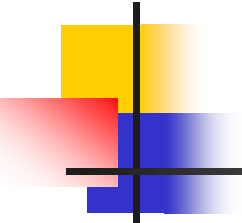


Build and run.

# Add Class Show



# Add Class Show



Generic C++ Class Wizard - Show\_from\_XML

Welcome to the Generic C++ Class Wizard

Class name:

.h file:  ...

.cpp file:  ...

Base class:

Access:  ▼

☐ Virtual destructor

☐ Inline

☐ Managed

Finish Cancel

```
#pragma once
#include <string>
```

# Show.h

```
class Show
{
public:
    struct Price_Schedule
    {
        std::string name;           // Price Schedule Name
        int price_level_1;          // Dollars
        int price_level_2;          // Dollars
        int price_level_3;          // Dollars
    };
    static const int MAX_PRICE_SCHEDULES = 10;

private:
    std::string name;              // Name of show
    int duration;                  // Minutes
    int setup_time;                // Minutes
    int takedown_time;             // Minutes

    Price_Schedule price_schedules[MAX_PRICE_SCHEDULES];
    int number_of_price_schedules;
```

# Show.h (continued)

public:

```
Show(void) {};
```

```
Show(std::string name_, int duration_, int setup_time_,  
      int takedown_time_);
```

```
void Add_Price_Schedule(Price_Schedule& ps);
```

```
std::string Name() const { return name; };
```

```
int Duration() const { return duration; };
```

```
int Setup_Time() const { return setup_time; };
```

```
int Takedown_Time() const { return takedown_time; };
```

```
int Number_of_Price_Schedules() const { return number_of_price_schedules;  
};
```

```
void Display() const;
```

private:

```
void Display_Price_Schedule(const Price_Schedule& ps) const;
```

```
};
```





# Show.cpp

---

```
#include <iostream>
#include <cassert>
#include "Show.h"

using namespace std;

Show::Show(string name_, int duration_, int setup_time_, int takedown_time_) :
    name(name_), duration(duration_), setup_time(setup_time_),
    takedown_time(takedown_time_), number_of_price_schedules(0)
{}

void Show::Add_Price_Schedule(Price_Schedule& ps)
{
    assert(number_of_price_schedules < MAX_PRICE_SCHEDULES);
    price_schedules[number_of_price_schedules++] = ps;
}
```

Continued on next slide.



# Show.cpp (continued)

---

```
void Show::Display() const
{
    cout << "Show: " << name << endl;
    cout << "duration: " << duration << " minutes\n";
    cout << "setup time: " << setup_time << " minutes\n";
    cout << "takedown time: " << takedown_time << " minutes\n";

    for (int i = 0; i < number_of_price_schedules; ++i)
    {
        cout << "Price schedule " << i+1 << endl;
        Display_Price_Schedule(price_schedules[i]);
        cout << endl;
    }
}

void Show::Display_Price_Schedule(const Price_Schedule& ps) const
{
    cout << ps.name << endl;
    cout << "Price Level 1: " << ps.price_level_1 << endl;
    cout << "Price Level 2: " << ps.price_level_2 << endl;
    cout << "Price Level 3: " << ps.price_level_3 << endl;
}
```

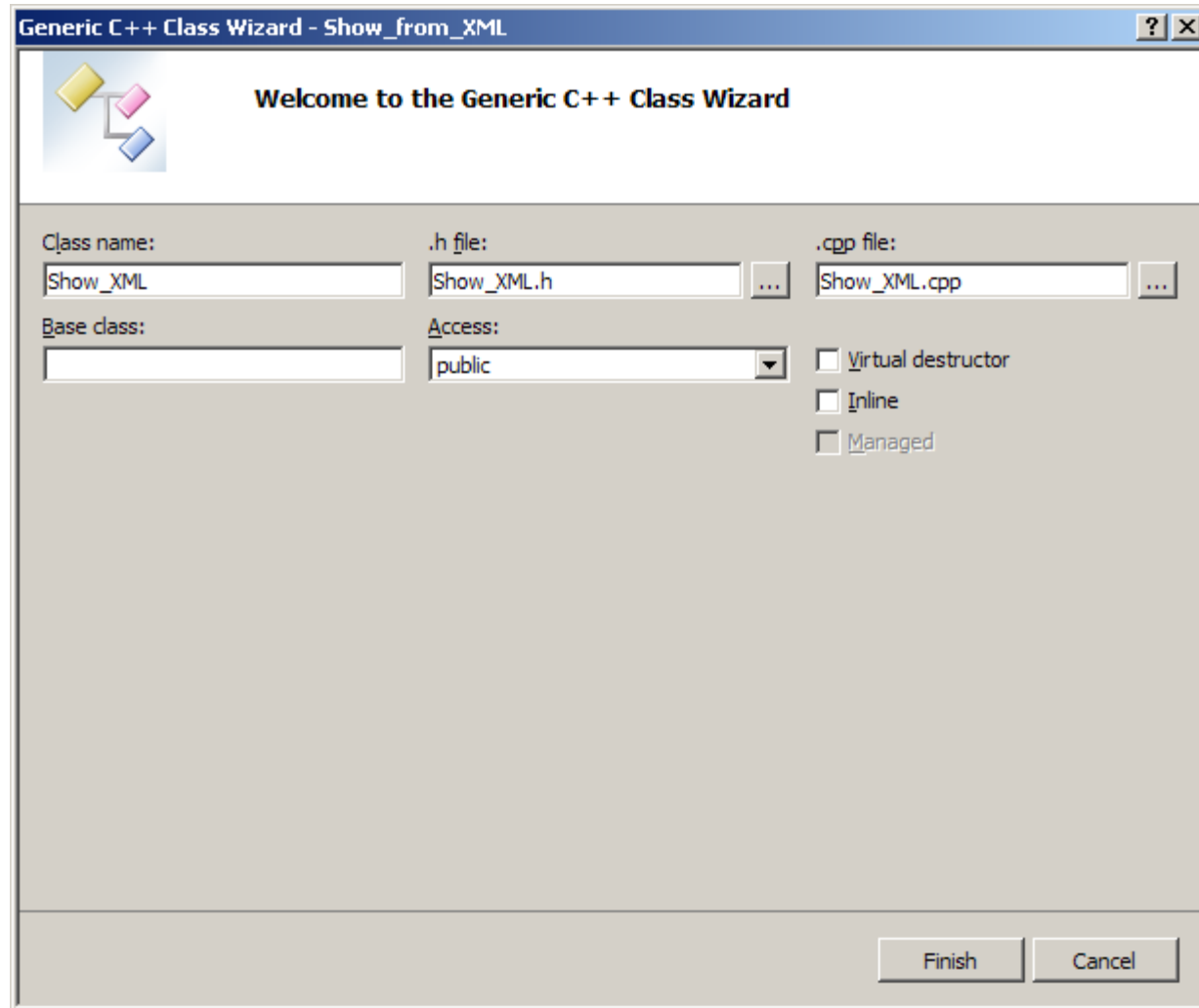


## Class Show\_XML

---

- We will put the XML specific code for Show into a separate *boudary* class, Show\_XML.
- Class Show\_XML knows about class Show and knows the structure of the XML file and the corresponding DOM.
- Creates a Show object from the DOM.
- Class Show does not know about Show\_XML.
  - No knowledge of the XML file or the DOM.
  - An entity class.

# Add Class Show\_XML



The image shows a 'Generic C++ Class Wizard' dialog box with the title 'Generic C++ Class Wizard - Show\_from\_XML'. It contains a 'Welcome to the Generic C++ Class Wizard' message and a wizard icon. The dialog has several input fields and checkboxes for configuring a new class.

**Class name:** Show\_XML

**.h file:** Show\_XML.h

**.cpp file:** Show\_XML.cpp

**Base class:** (empty)

**Access:** public

☐ Virtual destructor

☐ Inline

☐ Managed

**Buttons:** Finish, Cancel



# Show\_XML.h

---

```
#pragma once
#include "tinyxml.h"
#include "Show.h"

class Show_XML
{
public:
    static Show* Get_Show(TiXmlNode* show_node);

private:
    static Show::Price_Schedule
        Get_Price_Schedule(TiXmlNode* price_schedule_node);
};
```



# Show\_XML.cpp

---

```
#include <string>
#include "Show_XML.h"
#include "Show.h"
#include "tinyxml.h"
using namespace std;

Show* Show_XML::Get_Show(TiXmlNode* show_node)
{
    string name;
    string duration;
    string setup_time;
    string takedown_time;

    TiXmlNode* name_node = show_node->FirstChild();
    assert(name_node != 0);
    name = name_node->FirstChild()->Value();

    TiXmlNode* duration_node = name_node->NextSibling();
    assert(duration_node != 0);
    duration = duration_node->FirstChild()->Value();
```

Continued on next slide.



# Show\_XML.cpp (continued)

---

```
TiXmlNode* setup_time_node = duration_node->NextSibling();
assert(setup_time_node != 0);
setup_time = setup_time_node->FirstChild()->Value();

TiXmlNode* takedown_time_node = setup_time_node->NextSibling();
assert(takedown_time_node != 0);
takedown_time = takedown_time_node->FirstChild()->Value();

int int_duration = stoi(duration);
int int_setup_time = stoi(setup_time);
int int_takedown_time = stoi(takedown_time);

Show* show = new Show(name, int_duration, int_setup_time, int_takedown_time);

return show;
}
```



# Add to main.cpp

---

```
#include "Show.h"
#include "Show_XML.h"
...

string show_filename = "Show.xml";
TiXmlDocument doc(show_filename);

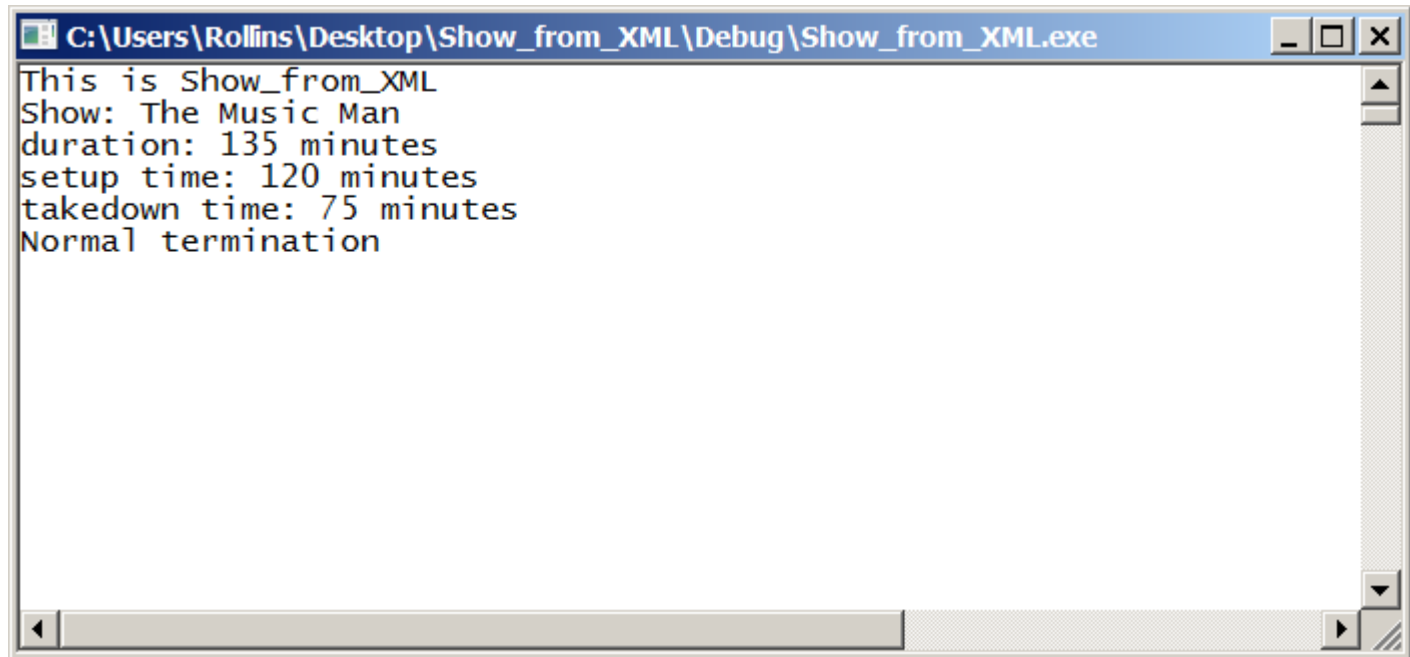
bool loadOkay = doc.LoadFile();
if (!loadOkay)
{
    cout << "Could not load file " << show_filename << endl;
    cout << "Error='" << doc.ErrorDesc() << "'. Exiting.\n";
    cin.get();
    exit( 1 );
}

TiXmlNode* show_node = doc.FirstChild("show");
assert(show_node != 0);
Show* show = Show_XML::Get_Show(show_node);

show->Display();
```



# Program in Action



```
C:\Users\Rollins\Desktop\Show_from_XML\Debug\Show_from_XML.exe
This is Show_from_XML
Show: The Music Man
duration: 135 minutes
setup time: 120 minutes
takedown time: 75 minutes
Normal termination
```



# Show\_XML.cpp

---

- At end of Get\_Show:

```
TiXmlNode* price_schedule_node = takedown_time_node->NextSibling();
int i = 0;
while ((price_schedule_node != 0) && (i < Show::MAX_PRICE_SCHEDULES))
{
    Show::Price_Schedule price_schedule =
        Get_Price_Schedule(price_schedule_node);
    price_schedule_node = price_schedule_node->NextSibling();
    show->Add_Price_Schedule(price_schedule);
}

return show;
}
```

# Show\_XML.cpp

```
Show::Price_Schedule
    Show_XML::Get_Price_Schedule(TiXmlNode* price_schedule_node)
{
    Show::Price_Schedule ps;

    TiXmlNode* name_node = price_schedule_node->FirstChild();
    assert(name_node != 0);
    ps.name = name_node->FirstChild()->Value();

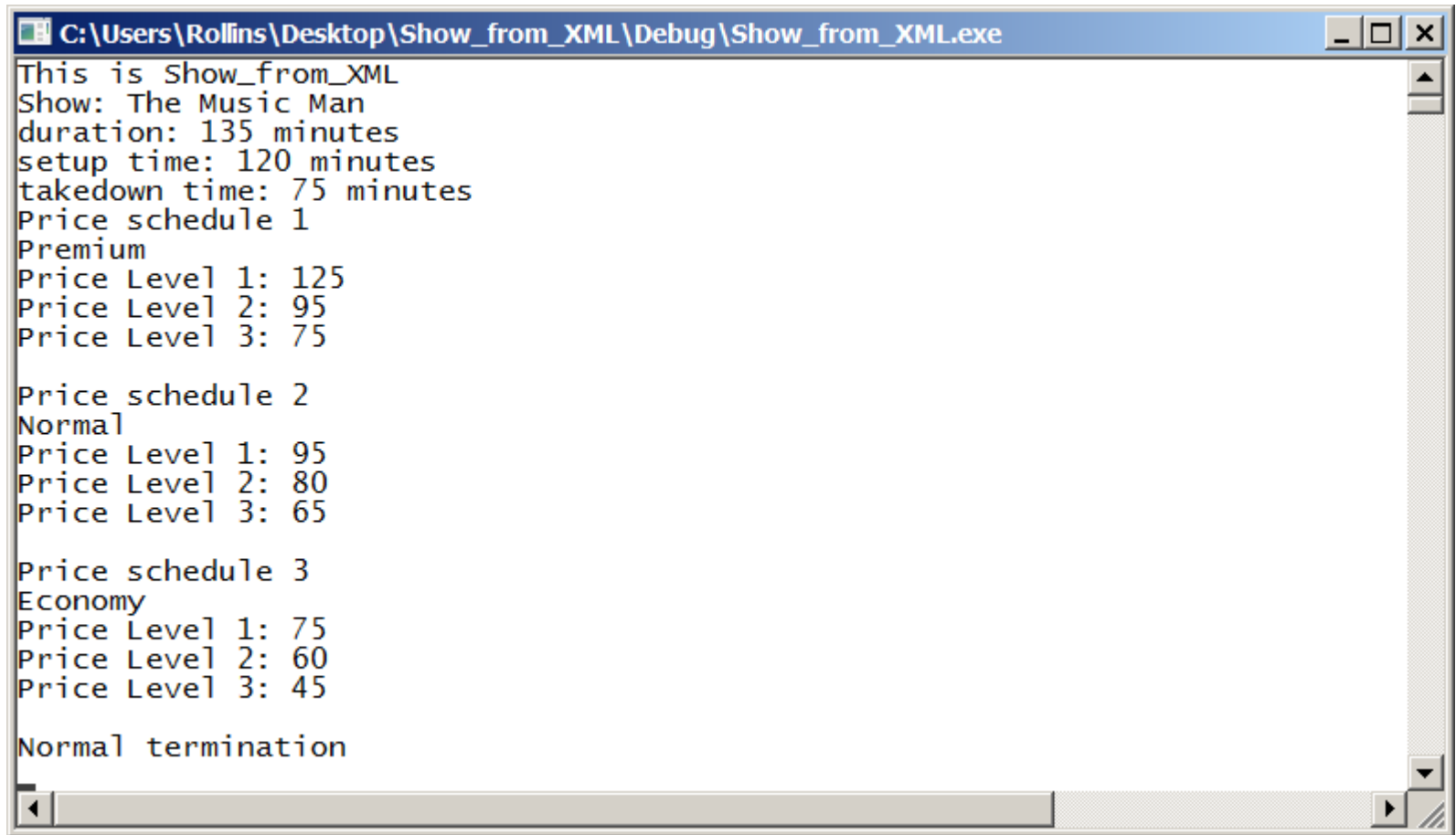
    TiXmlNode* price_level_1_node = name_node->NextSibling();
    assert(price_level_1_node != 0);
    ps.price_level_1 = stoi(price_level_1_node->FirstChild()->Value());

    TiXmlNode* price_level_2_node = price_level_1_node->NextSibling();
    assert(price_level_2_node != 0);
    ps.price_level_2 = stoi(price_level_2_node->FirstChild()->Value());

    TiXmlNode* price_level_3_node = price_level_2_node->NextSibling();
    assert(price_level_3_node != 0);
    ps.price_level_3 = stoi(price_level_3_node->FirstChild()->Value());

    return ps;
}
```

# Program in Action



```
C:\Users\Rollins\Desktop\Show_from_XML\Debug\Show_from_XML.exe
This is Show_from_XML
Show: The Music Man
duration: 135 minutes
setup time: 120 minutes
takedown time: 75 minutes
Price schedule 1
Premium
Price Level 1: 125
Price Level 2: 95
Price Level 3: 75

Price schedule 2
Normal
Price Level 1: 95
Price Level 2: 80
Price Level 3: 65

Price schedule 3
Economy
Price Level 1: 75
Price Level 2: 60
Price Level 3: 45

Normal termination
```