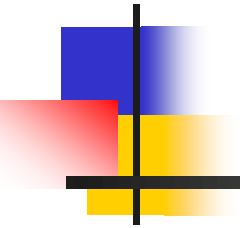


Project 7: Ticket_Booth_0

Using XML

Ticket Booth 0

- Base Level 0 of the real Ticket Booth program.

- Gets information for a single venue from an XML file.
- Instantiates a venue object.
  - See class diagram on a later slide.
- Displays results by calling the venue object's Display method.

Reuse Code

- Reuse classes from the previous projects as appropriate.
  - You will need to modify some of them.
    - Each class should have a Display method.
  - OK to use code from posted solutions.
- Reuse code from the class example, Test_Venue.
  - tinyxml should be used unchanged.
  - Modify boundary class Venue_from_Xml to meet the requirements of this project.
    - Instantiate objects rather than just displaying information.
    - Delete outputs when program is working

Program Specifications

- Program should use C++ strings rather than C strings.

- Seat should be a class.
  - A seat knows its row name, its seat number, and its section name.
  - Knows how to display itself.

Class Venue

- A Venue has a collection of Seat_Rows and a collection of Sections.

- Each Seat object will be in one Seat_Row and one Section

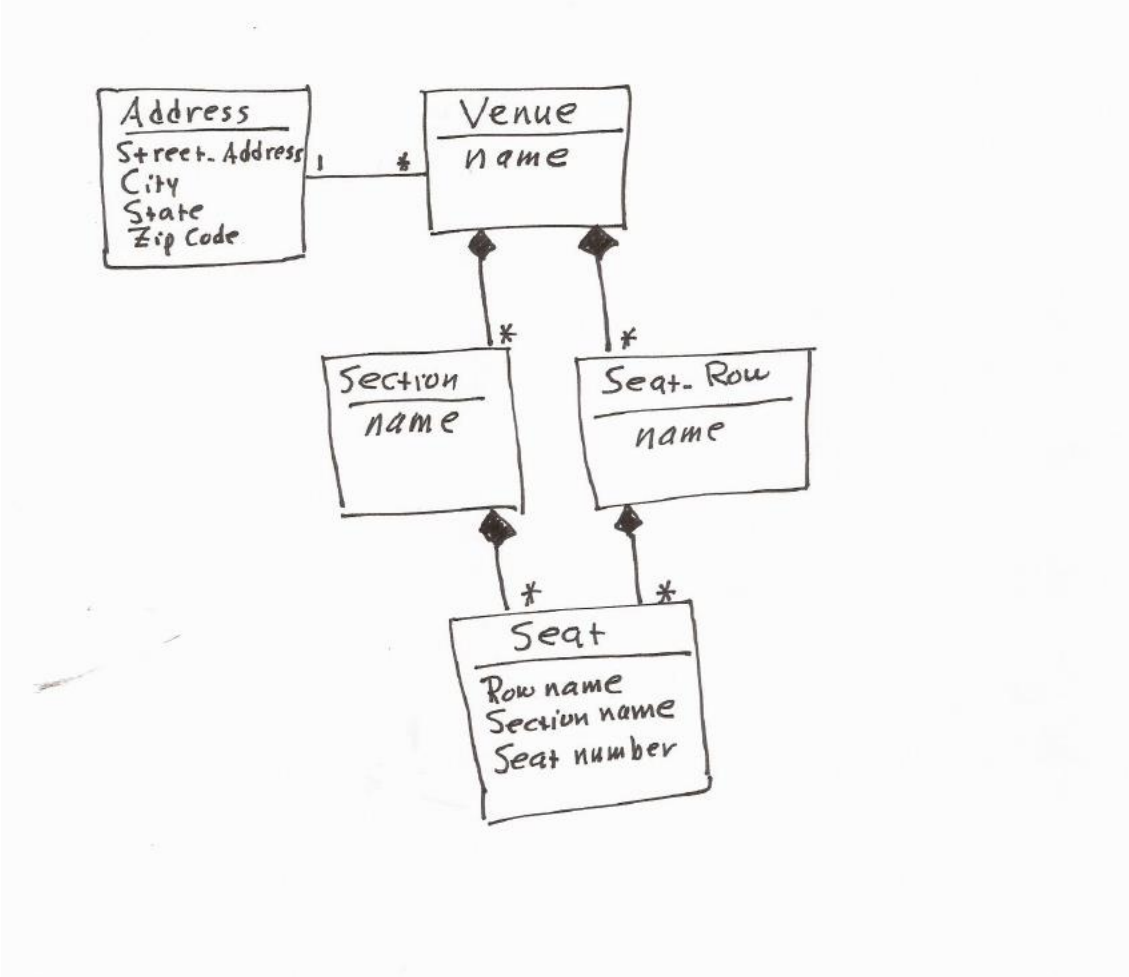
Class Seat_Row

- A Seat_Row object should have a collection of Seat objects.
  - Don't assume numbers are consecutive.
- Class Seat_Row will need an Add_Seat method.
  - Initialize Seat_Row with no seats.
  - Maximum of 1000 seats in a row.
- Class Seat_Row should have a Display method.

Class Section

- A Section object should have a collection of Seat objects.
  - Don't assume numbers are consecutive.
  - Don't assume all seats are in the same row.
- Class Section will need an Add_Seat method.
  - Initialize Section with no seats.
  - Maximum of 1000 seats in a row.
- Class Section should have a Display method.

Class Diagram

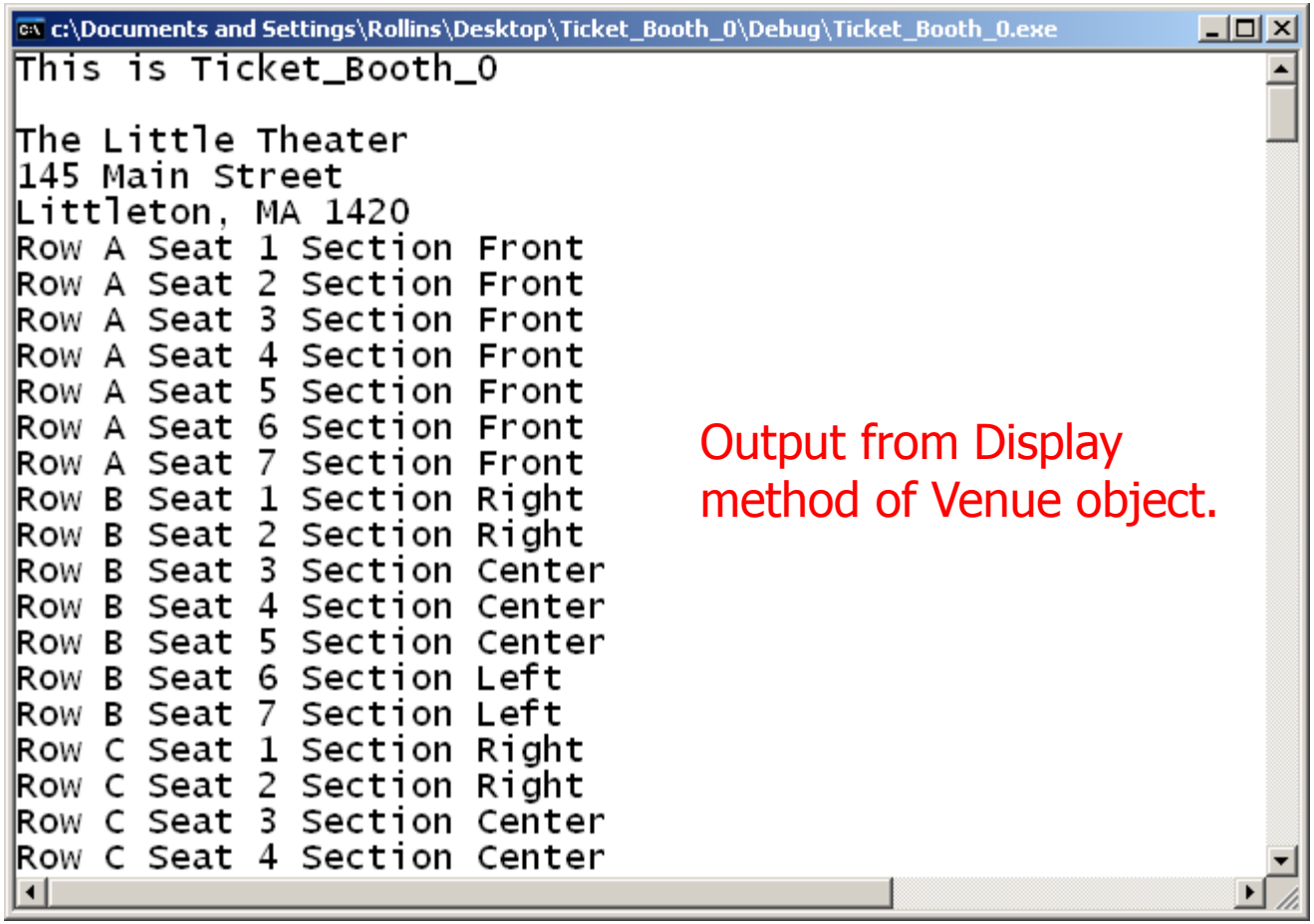


Sample Input File

http://www.csee.usf.edu/~turnerr/Object_Oriented_Design/Downloads/2016_03_01_XML/Venue.xml

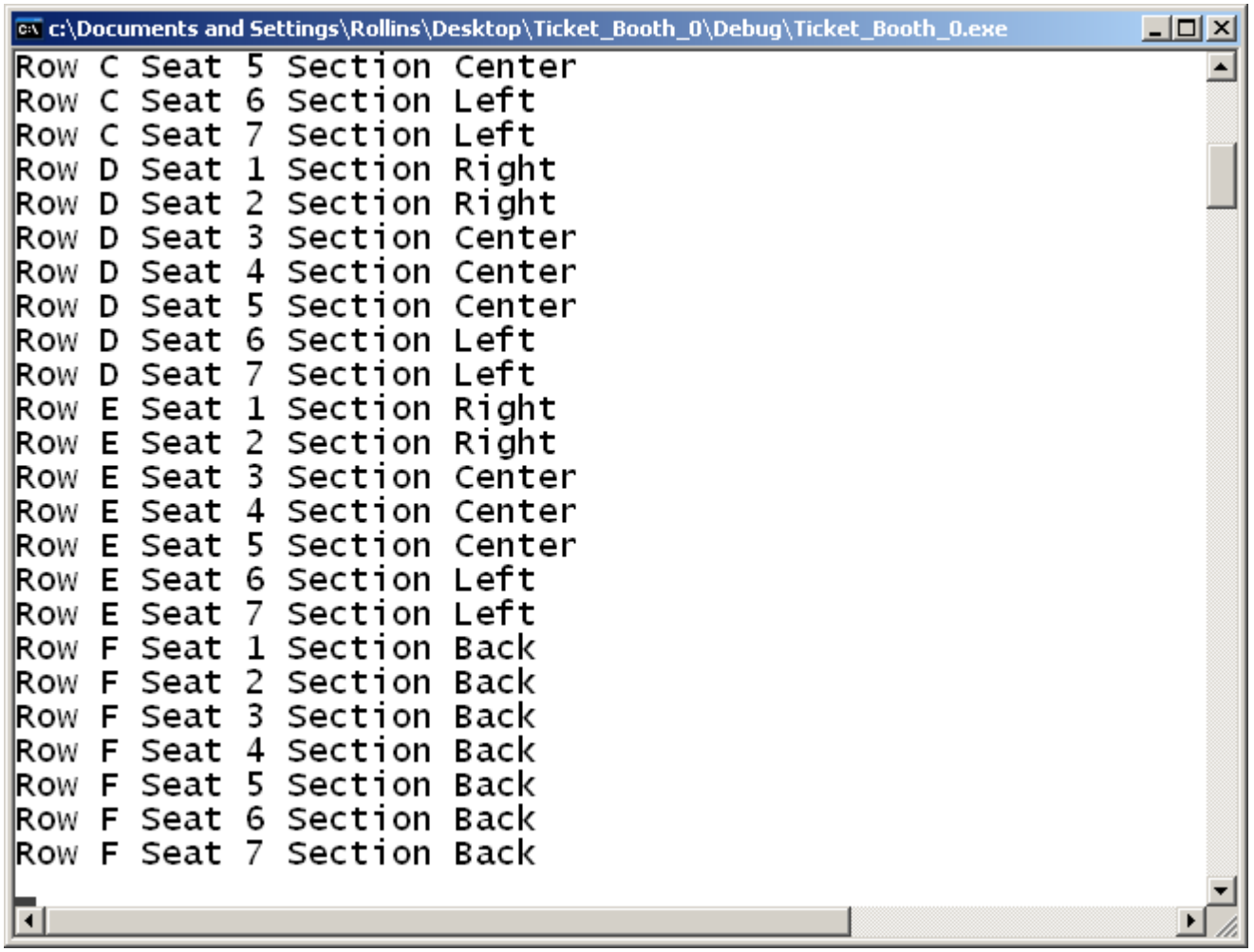
- Don't hard code information about the venue.
  - Other than maximum values.
- Everything that the program knows about the venue (beyond what is shown in the class diagram) should come from the XML file.
- The program should work with *any* valid XML file of this form.

Expected Output



Output from Display method of Venue object.

Expected Output (continued)



Ground Rules

This is a team project.

- Discuss the requirements.
- Divide up the work.
  - Report back and discuss your results.
- Record notes from your team meetings.
  - Distribute to all team members.
  - Submit in Canvas.

Ground Rules

- Do not share your work with other students outside your team.
  - Before or after submitting the project.
  - OK to *discuss* the project.
- Do not copy any other student's work.
  - Don't *look at* anyone else's program.
  - Don't let anyone look at your program.

Ground Rules

Except for code posted on the class web site

- Do not copy code from the Internet
  - or any other source.

- Write your own code.

Submission

- Assignment is due by 11:59 PM, Thursday, April 7.
- Deliverables:
  - Notes from team meetings.
  - Source code for the program.
    - Zipped project folder
- Put the zipped project folder and all other files into a folder and zip that folder for submission.
  - Submit a single .zip file.
- Use the Canvas Assignment to submit your work.
  - Only one submssion per team.
  - The team leader should do the submission.
  - Identify the team and members in a Canvas comment.