CS *251 Introduction:
Course Overview & Logistics

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Lesson

• Understand the course topics & logistics
  • Course philosophy
  • Course contents
  • Structure of the Lecture Material
  • Overview of the assignments & assessments
Course Philosophy
Course Philosophy

• While patterns & programming can be discussed abstractly, effective design & implementation practices are not best learned by generalities.

“Sitting & thinking” is not sufficient...
Course Philosophy

- While patterns & programming can be discussed abstractly, effective design & implementation practices are not best learned by generalities.
- Instead, it’s better to see how applying timeless software patterns & advanced object-oriented & generic design & programming techniques can help improve nontrivial programs.
While patterns & programming can be discussed abstractly, effective design & implementation practices are not best learned by generalities.

Instead, it’s better to see how applying timeless software patterns & advanced object-oriented & generic design & programming techniques can help improve nontrivial programs, e.g.,

- Easier to write & read;
- Easier to maintain & modify;
- More efficient & robust.

Course Philosophy

This course involves *lots* of hands-on software review, development, & testing.
Foundations for this Course Philosophy

- I’ve been programming C++ since 1986
Foundations for this Course Philosophy

- I’ve been programming C++ since 1986
- I was editor-in-chief of the C++ Report & have written books on C++ & patterns

Foundations for this Course Philosophy

• I’ve been programming C++ since 1986
• I was editor-in-chief of the C++ Report & have written books on C++ & patterns
• I’ve written millions of lines of widely-used open-source C++ software

See download.dre.vanderbilt.edu
Foundations for this Course Philosophy

- I’ve been programming C++ since 1986
- I was editor-in-chief of the C++ Report & have written books on C++ & patterns
- I’ve written millions of lines of widely-used open-source C++ software
- Taught more than 500,000 students in face-to-face & online courses
Summary of the Course Contents
Summary of Course Contents

• Topics important to developing & evolving quality C++ software
Summary of Course Contents

- Topics important to developing & evolving quality C++ software
- Reuse of patterns & software components
Summary of Course Contents

- Topics important to developing & evolving quality C++ software
  - Reuse of patterns & software components
- Developing, documenting, testing, & applying reusable classes & frameworks
Summary of Course Contents

- Patterns will be taught via an extended case study
Summary of Course Contents

- Patterns will be taught via an extended case study
- Provide good role models for software designs

C++
Stack
Queue

ExpressionTree
ComponentNode
Composite
UnaryNode
LeafNode
Composite
BinaryNode
...
• Patterns will be taught via an extended case study
• Provide good role models for software designs
• Clearly articulate design tradeoffs
Summary of Course Contents

- Object-oriented & generic design & programming techniques will be taught via many examples

See github.com/douglascraigschmidt/CPlusPlus
Summary of Course Contents

- Object-oriented & generic design & programming techniques will be taught via many examples
- e.g., show how to build software architectures that minimize dependencies & coupling between components

**Observer pattern**

```plaintext
for all observers in observerList do
    observer.update()
```

```plaintext
s->getData()
```
Summary of Course Contents

• I assume you know some C++
  • i.e., equivalent to what’s covered in CS 2201
Summary of Course Contents

• I assume you know some C++
  • i.e., equivalent to what’s covered in CS 2201

No matter what you know about C++, you’ll learn a lot more!
Structure of the Lecture Material
Structure of the Lecture Material

- This course has four main sections

<table>
<thead>
<tr>
<th>Section</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++ Review &amp; Overview</td>
<td>• Overview of basic object-oriented &amp; generic programming features</td>
</tr>
<tr>
<td></td>
<td>• Overview of the Clion IDE</td>
</tr>
<tr>
<td>Advanced C++ Programming</td>
<td>• C++11 &amp; C++14 features</td>
</tr>
<tr>
<td>Standard Template Library (STL)</td>
<td>• Containers, iterators, algorithms</td>
</tr>
<tr>
<td>Design Patterns</td>
<td>• Expression tree case study</td>
</tr>
</tbody>
</table>
Structure of the Lecture Material

- This course has four main sections
  - Each Section is composed of Modules
Structure of the Lecture Material

• This course has four main sections
  • Each Section is composed of Modules
  • Each Module is composed of Parts
Structure of the Lecture Material

• This course has four main sections
  • Each Section is composed of Modules
  • Each Module is composed of Parts
  • Each Part is a single lecture
This course has four main sections

- Each Section is composed of Modules
- Each Module is composed of Parts
- Each Part is a single lecture
- Each Part is composed of segments

Screencasts of each lecture “Part” & slides will be recorded & uploaded to www.dre.vanderbilt.edu/~schmidt/cs251
Structure of the Lecture Material

• This course has four main sections
• There will be a weekly quiz on material covered in the lectures

I recommend studying for quizzes by watching screencasts of lectures at [www.dre.vanderbilt.edu/~schmidt/cs251](http://www.dre.vanderbilt.edu/~schmidt/cs251)
Structure of the Lecture Material

• This course has four main sections
• There will be a weekly quiz on material covered in the lectures
  • 1st quiz will be on Friday using Brightspace
Structure of the Lecture Material

• This course has four main sections

• There will be a weekly quiz on material covered in the lectures
  - 1st quiz will be on Friday using Brightspace

• Graded quizzes will be available by the start of the next class
Structure of the Lecture Material

- This course has four main sections
- There will be a weekly quiz on material covered in the lectures
- There will be a cumulative final exam that covers all the lectures

Final exam time is 9am to noon, final day of the class
Structure of the Lecture Material

• This course has four main sections
• There will be a weekly quiz on material covered in the lectures
• There will be a cumulative final exam that covers all the lectures
• The relative weighting* of each portion of the course is:
  • 45% Quizzes
  • 40% Programming projects
  • 10% Final exam
  • 05% Participation
* Relative weighting may change
Overview of the Assignments & Assessments
Overview of Assignments & Assessments

• Programming assignments should be written in C++ using Clion 2020.1

```cpp
int main() {
    try {
        // Multiple stacks that are created automatically.
        stack<int> s1(size: 10), s2(size: 100);
        int item = 0;

        while (!s1.full())
            s1.push(item++);

        while (!s1.empty()) {
            cout << "top item = " << s1.top() << endl;
            s1.pop();
        }

        s1 = s2; // No aliasing problem with copy assignment
        // s1.top_ = 10; // Access problem caught at compile-time!
        // Termination is handled automatically.
    } catch (std::out_of_range &ex) {
        cout << "caught out of range exception" << endl;
    }
}
```

See [github.com/douglascraigschmidt/CS251/wiki/Installing-Software](https://github.com/douglascraigschmidt/CS251/wiki/Installing-Software)
Overview of Assignments & Assessments

• Programming assignments should be written in C++ using Clion 2020.1

• All source code for assignments & examples available at GitHub

Go to GitHub at github.com/douglascraigschmidt/CS251
Overview of Assignments & Assessments

- Programming assignments should be written in C++ using Clion 2020.1
- All source code for assignments & examples available at GitHub
- You will need to learn how to use GitLab et al.

See github.com/douglascraigschmidt/CS251/wiki/Installing-Software
Overview of Assignments & Assessments

• Programming assignments should be written in C++ using Clion 2020.1

• All source code for assignments & examples available at GitHub
  • You will need to learn how to use GitLab et al.

• Be prepared to update your repositories multiple times

“If you don’t like change, you’re going to like irrelevance even less.”
Overview of Assignments & Assessments

• Programming assignments should be written in C++ using Clion 2020.1
• All source code for assignments & examples available at GitHub
• The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming

See github.com/douglasraigschmidt/CS251/tree/master/assignments
Overview of Assignments & Assessments

- Programming assignments should be written in C++ using Clion 2020.1
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming
  - Assignments & examples will be released throughout the course

See [github.com/douglasraigschmidt/CS251/tree/master/assignments](https://github.com/douglasraigschmidt/CS251/tree/master/assignments)
Overview of Assignments & Assessments

- Programming assignments should be written in C++ using Clion 2020.1
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming
- Assessments will be done via reviews by course staff

See [www.dre.vanderbilt.edu/~schmidt/cs251/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs251/assignments.html)
Overview of Assignments & Assessments

• Programming assignments should be written in C++ using Clion 2020.1
• All source code for assignments & examples available at GitHub
• The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming
• Assessments will be done via reviews by course staff
Overview of Assignments & Assessments

• Programming assignments should be written in C++ using Clion 2020.1
• All source code for assignments & examples available at GitHub
• The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming
• Assessments will be done via reviews by course staff
  • Assignments & reviews *must* be submitted on time or you will receive a 0

See [www.dre.vanderbilt.edu/~schmidt/cs251/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs251/assignments.html)
Programming assignments should be written in C++ using Clion 2020.1

All source code for assignments & examples available at GitHub

The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming

Assessments will be done via reviews by course staff

Assignments & reviews must be submitted on time or you will receive a 0

Work must be your own

www.vanderbilt.edu/student_handbook/the-honor-system#statement-of-the-honor-code
Overview of Assignments & Assessments

- Programming assignments should be written in C++ using Clion 2020.1
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming
- Assessments will be done via reviews by course staff
- Assessment criteria

<table>
<thead>
<tr>
<th>Assessment Category</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution correctness</td>
<td>40%</td>
</tr>
<tr>
<td>Structure (e.g., modularization, information hiding, etc.)</td>
<td>30%</td>
</tr>
<tr>
<td>Insightful programming (e.g., developing reusable class components, etc.)</td>
<td>10%</td>
</tr>
<tr>
<td>Consistent style (e.g., capitalization, indenting, etc.)</td>
<td>10%</td>
</tr>
<tr>
<td>Appropriate commenting style</td>
<td>10%</td>
</tr>
</tbody>
</table>
Summary
Summary

• You will get out of this course what you put into it
Summary

• You will get out of this course what you put into it
• Be prepared to work hard

HARD WORK

“Human Felicity is produc'd not so much by great Pieces of good Fortune that seldom happen, as by little Advantages that occur every Day” - Benjamin Franklin
Summary

• You will get out of this course what you put into it
• Be prepared to work hard
• Do *not* miss deadlines...
Summary

• You will get out of this course what you put into it
• Be prepared to work hard
• Do *not* miss deadlines...
• Participate

See piazza.com/vanderbilt/summer2020/cs3251
Summary

- You will get out of this course what you put into it
- Be prepared to work hard
- Do not miss deadlines...
- Participate
- Avail yourself of available resources

See www.dre.vanderbilt.edu/~schmidt/cs251
You will get out of this course what you put into it
• Be prepared to work hard
• Do not miss deadlines...
• Participate
• Avail yourself of available resources

Please resist the urge to email me directly!
• You will get out of this course what you put into it
• Be prepared to work hard
• Do *not* miss deadlines...
• Participate
• Avail yourself of available resources
• There are abundant opportunities!