

# **CS \*251 Introduction: Course Overview & Logistics**

**Douglas C. Schmidt**

**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

**[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)**



**Professor of Computer Science**

**Institute for Software  
Integrated Systems**

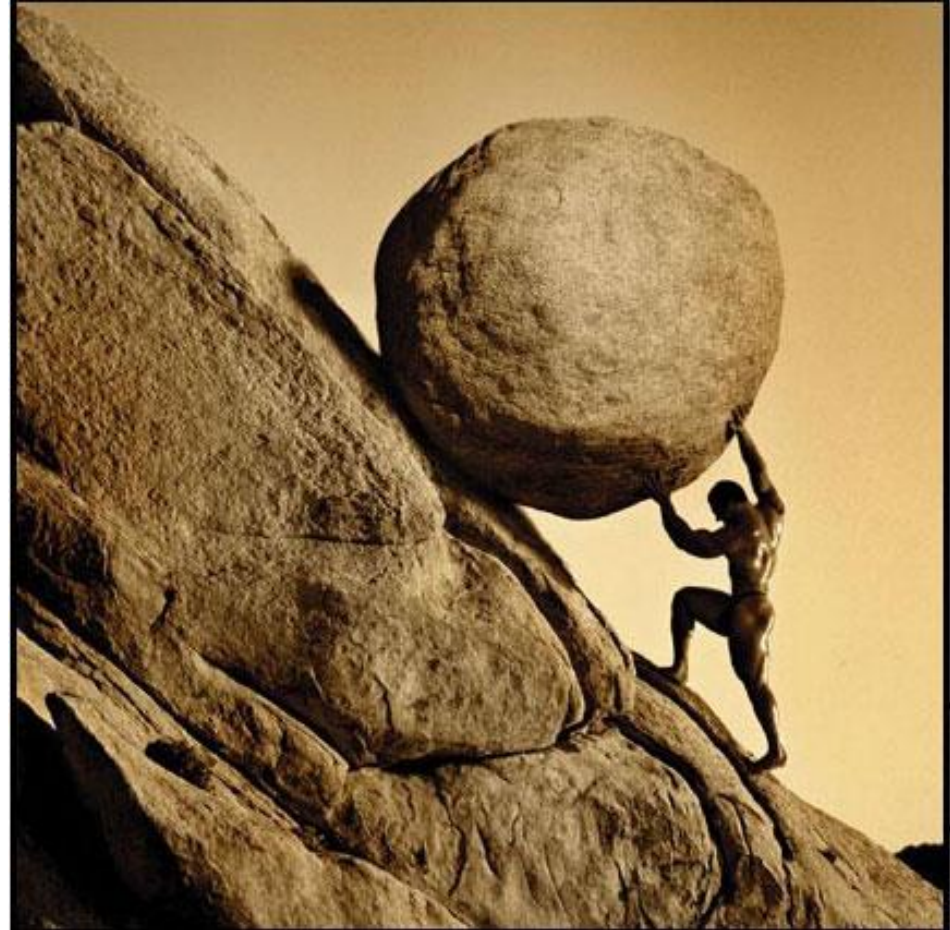
**Vanderbilt University  
Nashville, Tennessee, USA**



# 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



# 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, e.g.,
  - Easier to write & read;
  - Easier to maintain & modify;
  - More efficient & robust.



---

This course involves *lots* of hands-on software review, development, & testing

# Foundations for this Course Philosophy

---

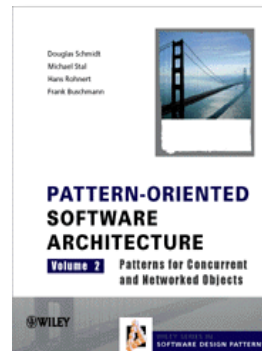
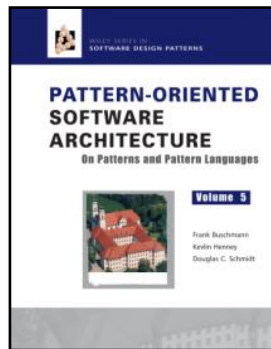
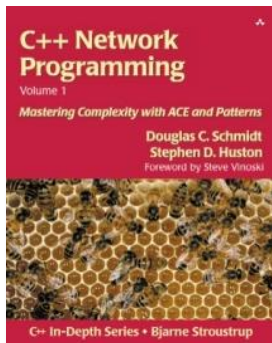
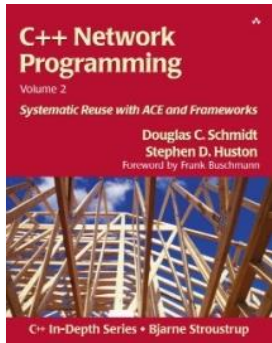
- I've been programming C++ since 1986





# 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



See [www.dre.vanderbilt.edu/~schmidt/CV.html](http://www.dre.vanderbilt.edu/~schmidt/CV.html)  
& [en.wikipedia.org/wiki/C++\\_Report](http://en.wikipedia.org/wiki/C++_Report)



# 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

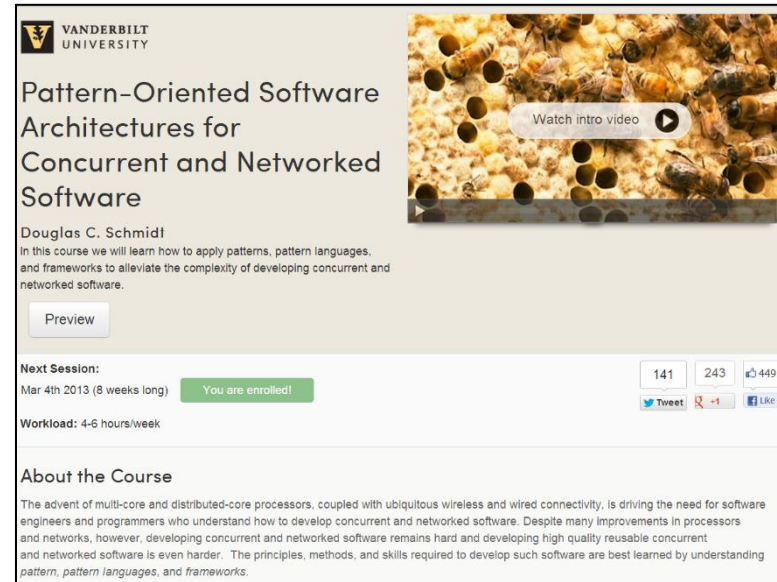


---

See [download.dre.vanderbilt.edu](http://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



VANDERBILT UNIVERSITY

## Pattern-Oriented Software Architectures for Concurrent and Networked Software

Douglas C. Schmidt

In this course we will learn how to apply patterns, pattern languages, and frameworks to alleviate the complexity of developing concurrent and networked software.

[Preview](#)

**Next Session:**  
Mar 4th 2013 (8 weeks long) [You are enrolled!](#)

**Workload:** 4-6 hours/week

**About the Course**

The advent of multi-core and distributed-core processors, coupled with ubiquitous wireless and wired connectivity, is driving the need for software engineers and programmers who understand how to develop concurrent and networked software. Despite many improvements in processors and networks, however, developing concurrent and networked software remains hard and developing high quality reusable concurrent and networked software is even harder. The principles, methods, and skills required to develop such software are best learned by understanding *pattern*, *pattern languages*, and *frameworks*.



## Android App Development Specialization

Launch Your Android App Development Career. Master the knowledge and skills necessary to develop maintainable mobile computing apps

See [www.dre.vanderbilt.edu/~schmidt/DigitalLearning](http://www.dre.vanderbilt.edu/~schmidt/DigitalLearning)

---

# 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

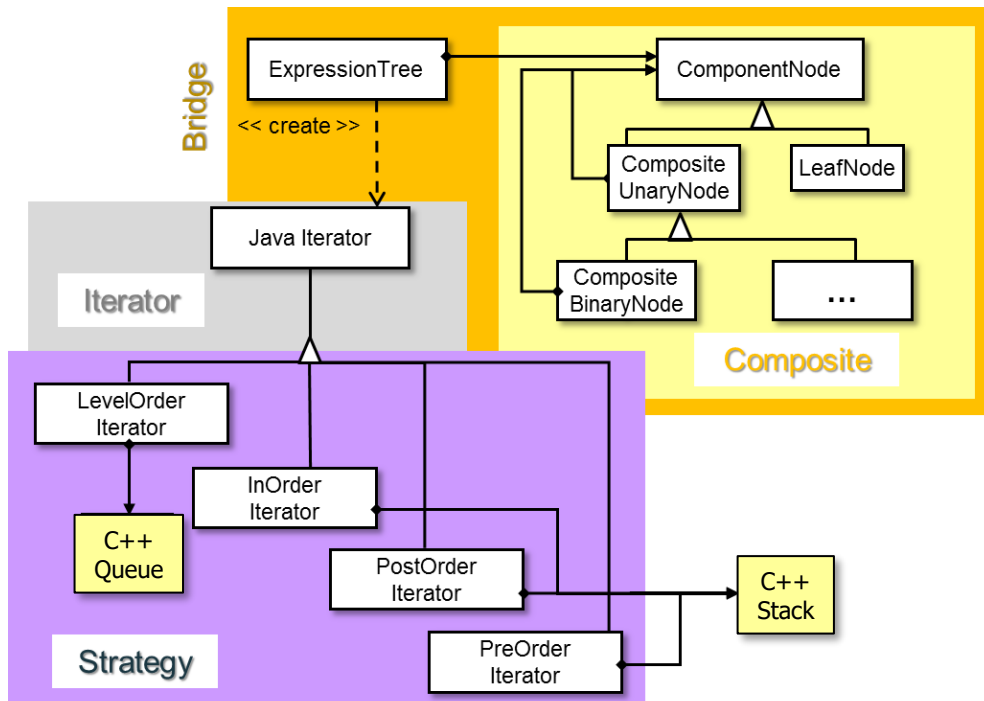
---

- 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



# Summary of Course Contents

- 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



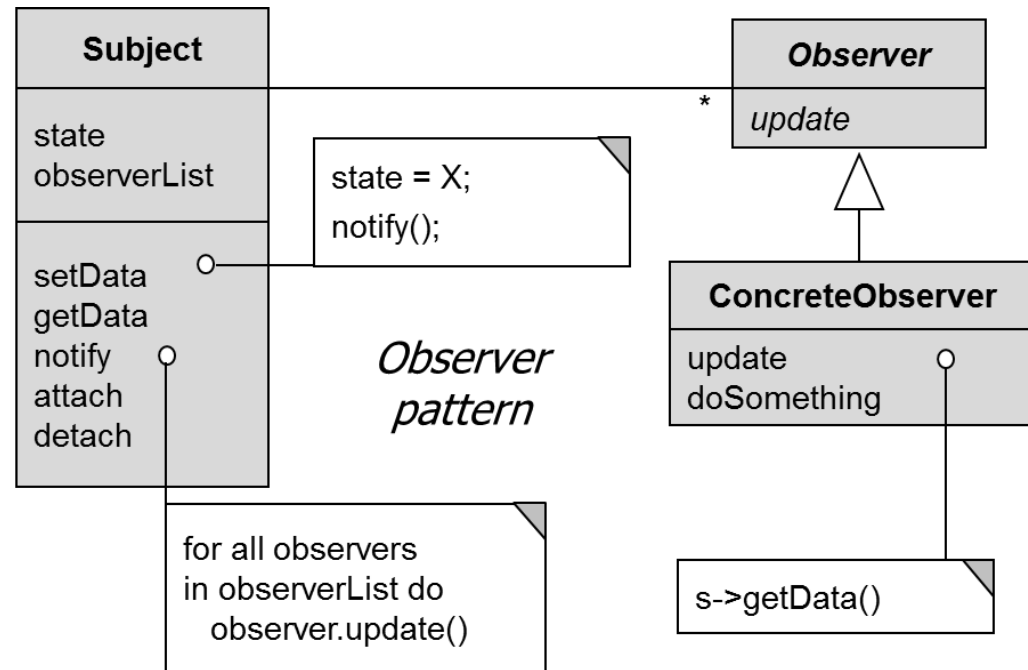
---

See [github.com/douglasraigschmidt/CPlusPlus](https://github.com/douglasraigschmidt/CPlusPlus)



# Summary of Course Contents

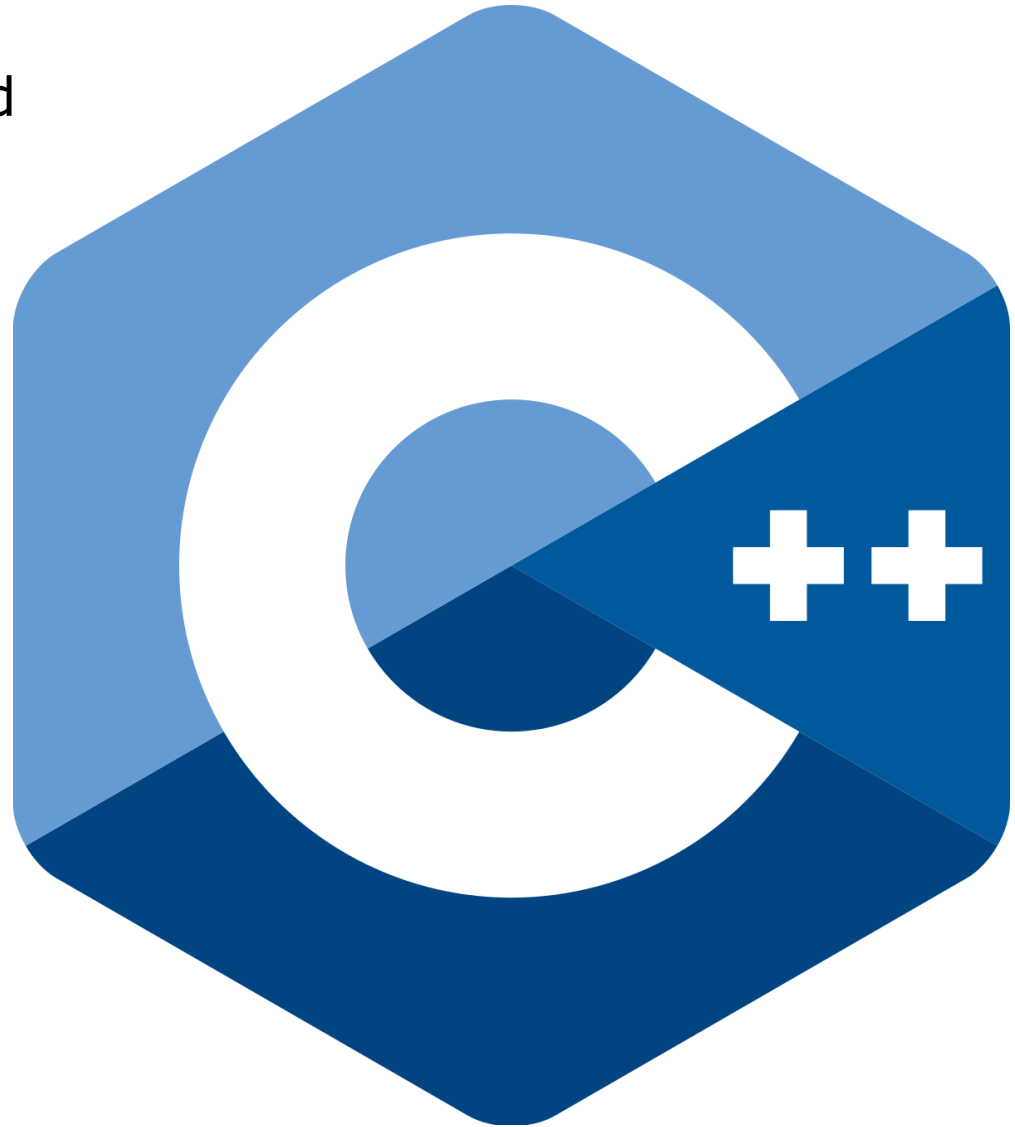
- Object-oriented & generic design & programming techniques will be taught via many examples
- e.g., show how to build software architectures that minimize dependencies & coupling between components



# 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

Section	Topics
C++ Review & Overview	<ul style="list-style-type: none"><li>• Overview of basic object-oriented &amp; generic programming features</li><li>• Overview of the Clion IDE</li></ul>
Advanced C++ Programming	<ul style="list-style-type: none"><li>• C++11 &amp; C++14 features</li></ul>
Standard Template Library (STL)	<ul style="list-style-type: none"><li>• Containers, iterators, algorithms</li></ul>
Design Patterns	<ul style="list-style-type: none"><li>• Expression tree case study</li></ul>



# 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



# 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
    - Each Part is composed of segments



Screencasts of each lecture "Part" & slides will be recorded & uploaded to [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



---

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
  - 1<sup>st</sup> 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
  - 1<sup>st</sup> 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

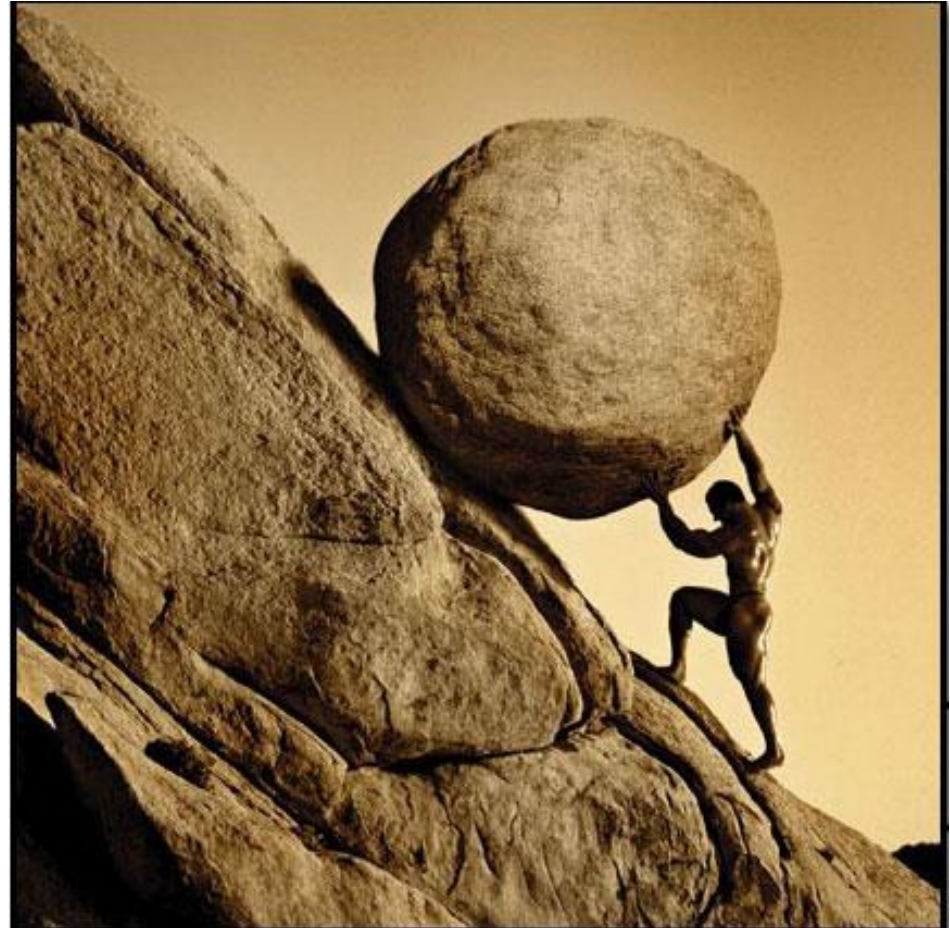


---

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

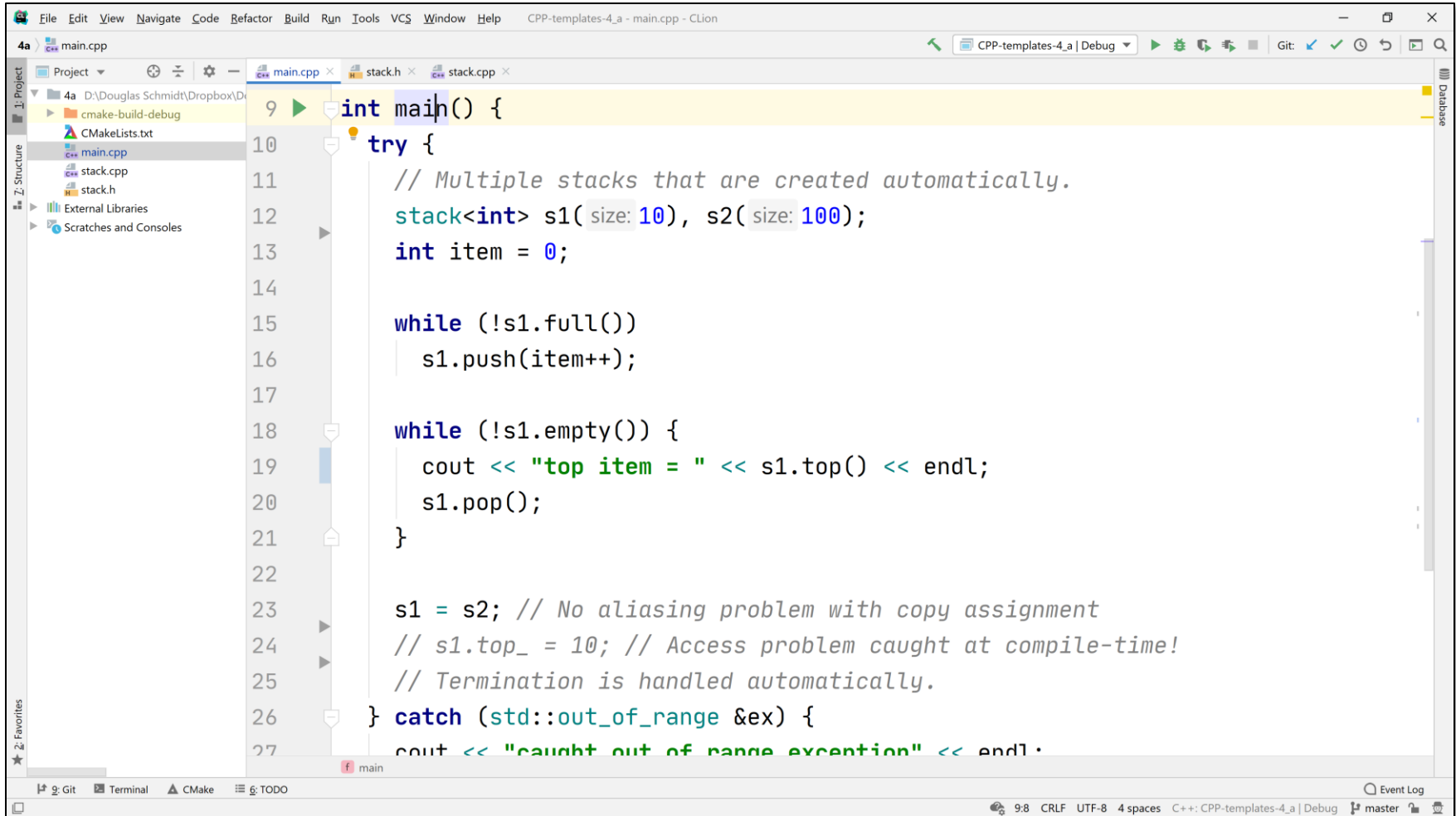


---

# Overview of the Assignments & Assessments

# Overview of Assignments & Assessments

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



```
1  int main() {
2      try {
3          // Multiple stacks that are created automatically.
4          stack<int> s1( size: 10), s2( size: 100);
5          int item = 0;
6
7          while (!s1.full())
8              s1.push(item++);
9
10         while (!s1.empty()) {
11             cout << "top item = " << s1.top() << endl;
12             s1.pop();
13         }
14
15         s1 = s2; // No aliasing problem with copy assignment
16         // s1.top_ = 10; // Access problem caught at compile-time!
17         // Termination is handled automatically.
18     } catch (std::out_of_range &ex) {
19         cout << "caught out of range exception" << endl;
20     }
```

See [github.com/douglasraigschmidt/CS251/wiki/Installing-Software](https://github.com/douglasraigschmidt/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

The screenshot shows the GitHub repository page for `douglasraigschmidt / CS251`. At the top, there are buttons for `Unwatch` (1), `Star` (2), and `Fork` (0). Below this, a description states: "This contains the source code examples and programming assignments for my CS 251 class — Edit". A summary bar shows `3 commits`, `1 branch`, `0 releases`, and `1 contributor`. The main content area shows a commit history table with the following entries:

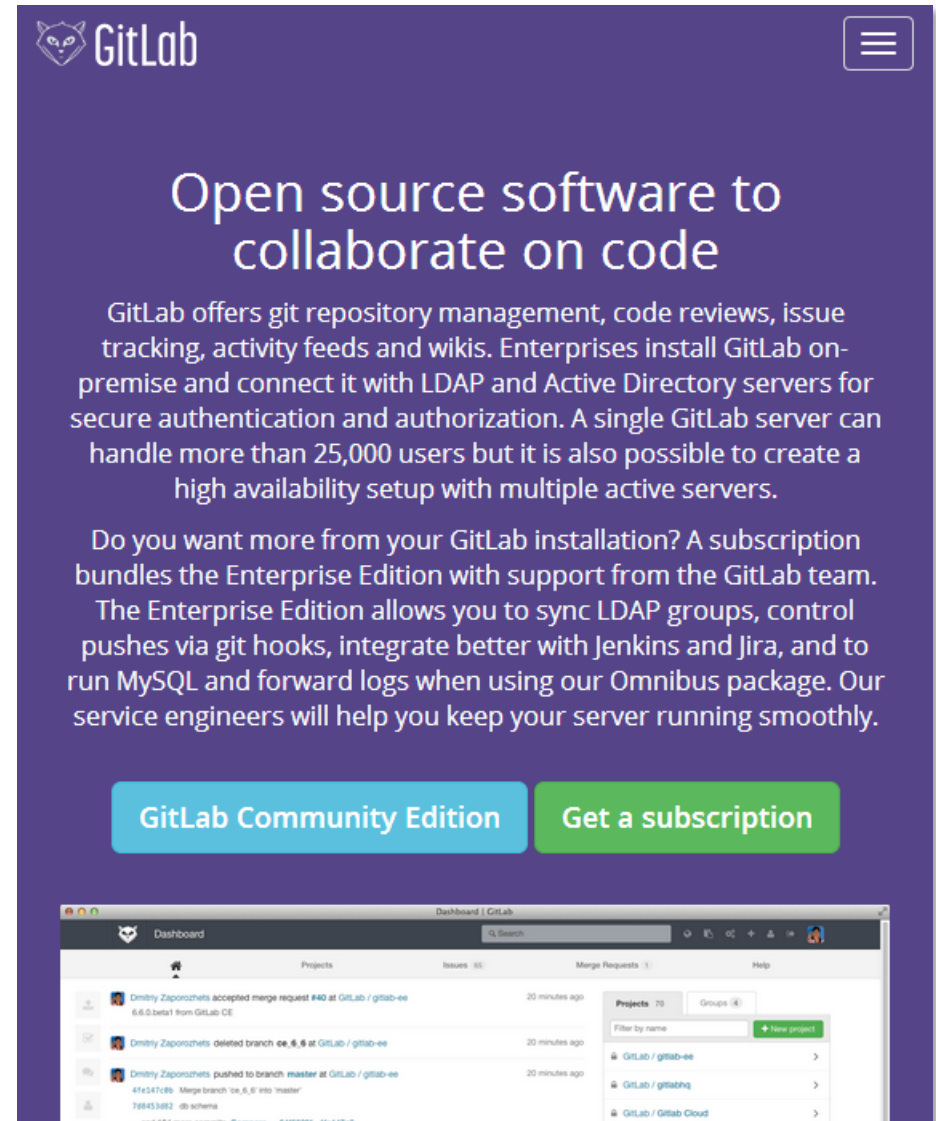
Commit Message	Time Ago
First pass.	14 seconds ago
assignments/assignment1	14 seconds ago
ex/ExpressionTree	9 minutes ago
.gitignore	9 minutes ago
README.md	5 days ago

Below the commit history is a section for `README.md` with the title **CS251** and the same description: "This contains the source code examples and programming assignments for my CS 251 class". On the right sidebar, there are links for `Code`, `Issues` (0), `Pull Requests` (0), `Wiki`, `Pulse`, `Graphs`, and `Settings`. At the bottom of the sidebar, there is an `SSH clone URL` field with the value `git@github.com:doug:` and buttons for `Clone in Desktop` and `Download ZIP`.

Go to GitHub at [github.com/douglasraigschmidt/CS251](https://github.com/douglasraigschmidt/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.



The image shows the GitLab landing page and a screenshot of the GitLab dashboard. The landing page has a purple background with the GitLab logo (a cat face) and the text "Open source software to collaborate on code". It describes GitLab's features: git repository management, code reviews, issue tracking, activity feeds, and wikis. It mentions that enterprises install GitLab on-premise and connect it with LDAP and Active Directory servers for secure authentication and authorization. A single GitLab server can handle more than 25,000 users but it is also possible to create a high availability setup with multiple active servers. It also promotes the Enterprise Edition with support from the GitLab team, allowing for LDAP group sync, Jenkins and Jira integration, and MySQL support. At the bottom of the landing page are two buttons: "GitLab Community Edition" (blue) and "Get a subscription" (green).

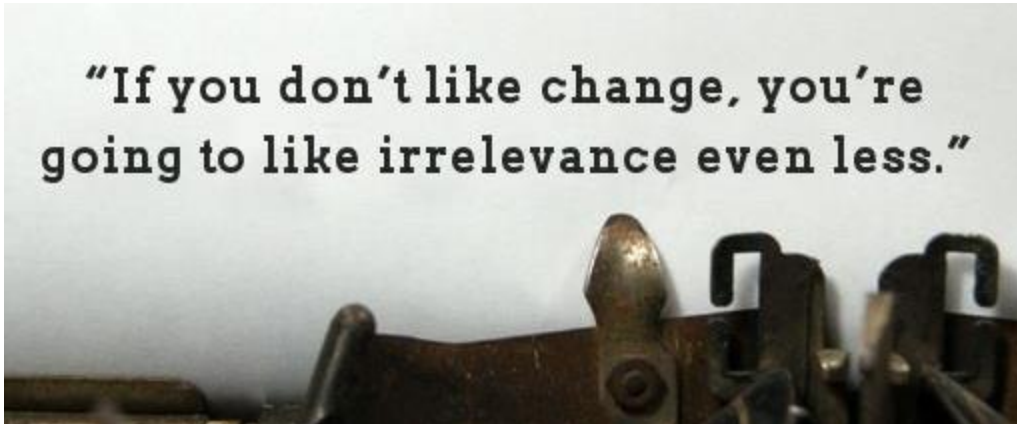
The screenshot below shows the GitLab dashboard. It has a dark header with the GitLab logo and a search bar. The main content area is divided into sections: "Projects" (70), "Issues" (55), "Merge Requests" (1), and "Help". The "Projects" section lists several projects, including "GitLab / gitlab-ee", "GitLab / gitlabhq", and "GitLab / GitLab Cloud". The "Issues" section shows a list of issues, including "Omri Zaporozhets accepted merge request #40 at GitLab / gitlab-ee" and "Omri Zaporozhets deleted branch ce\_6.6 at GitLab / gitlab-ee". The "Merge Requests" section shows a list of merge requests, including "Omri Zaporozhets pushed to branch master at GitLab / gitlab-ee" and "Omri Zaporozhets Merge branch 'ce\_6.6' into 'master'".

See [github.com/douglasraigschmidt/CS251/wiki/Installing-Software](https://github.com/douglasraigschmidt/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





# 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](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
  - 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

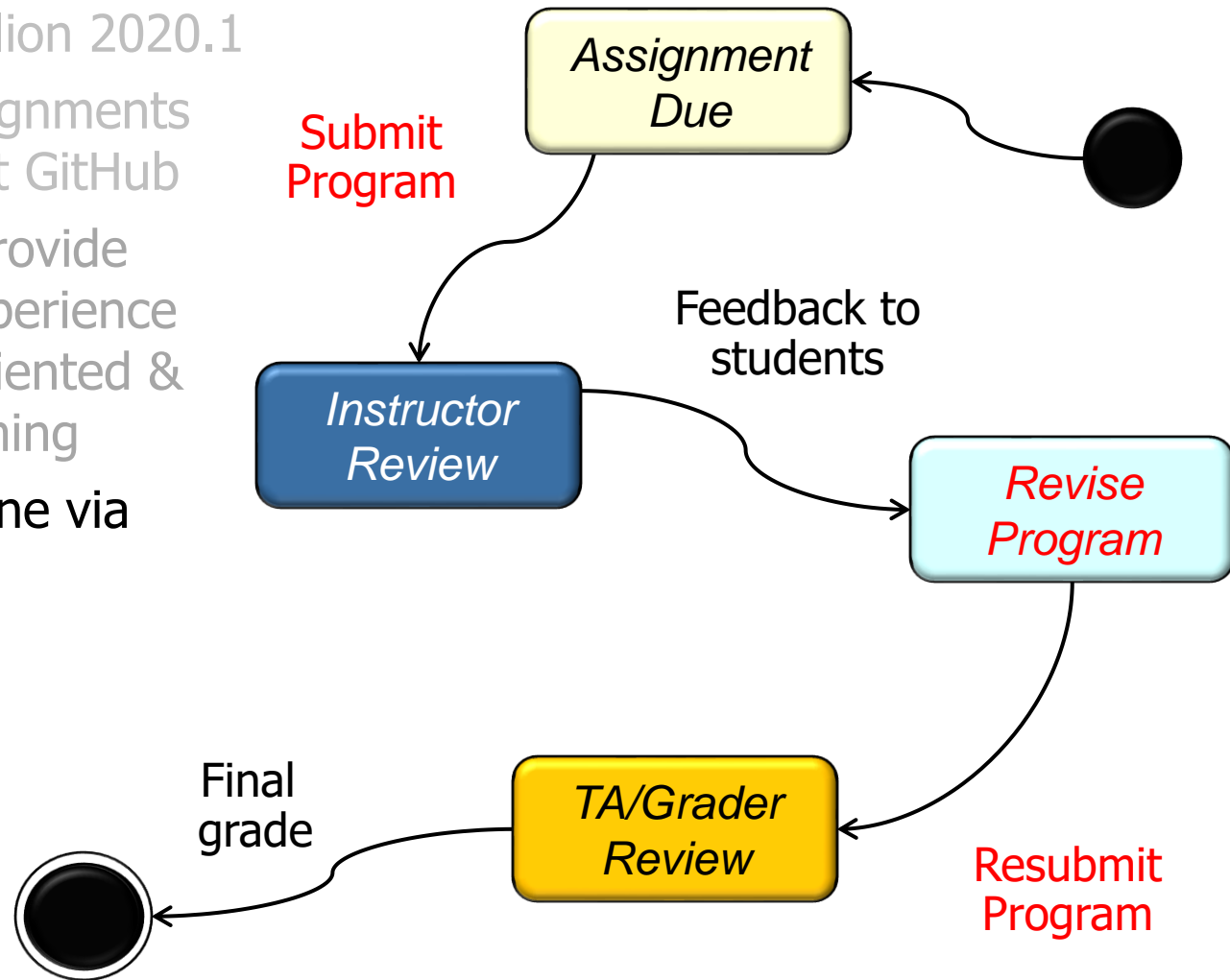


---

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

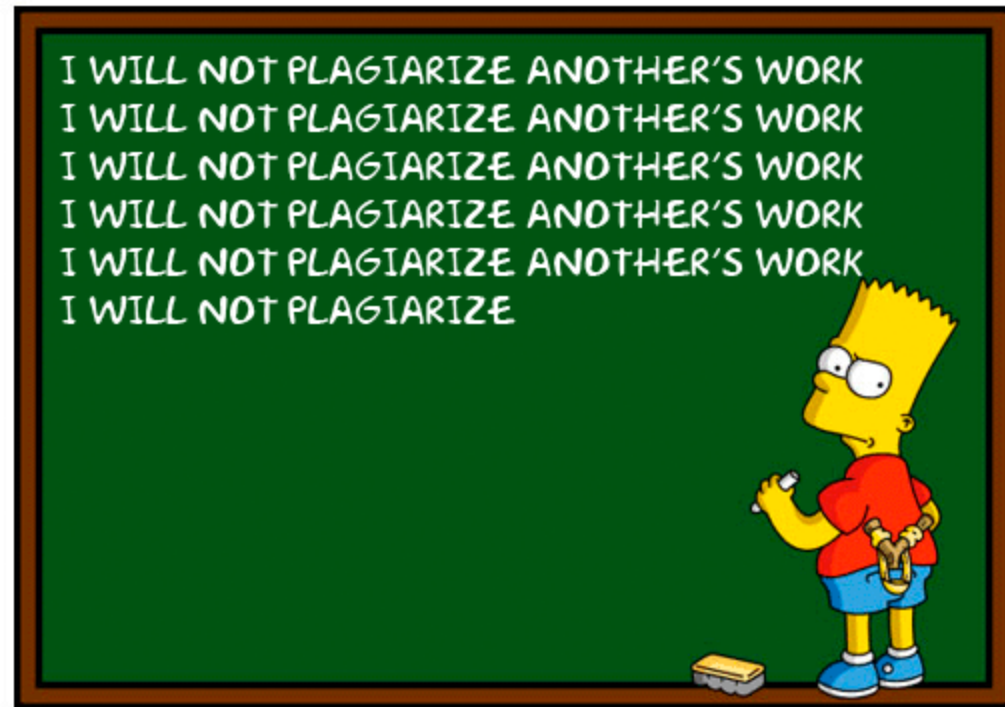


---

See [www.dre.vanderbilt.edu/~schmidt/cs251/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs251/assignments.html)

# 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
  - Work *must* be your own



# 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

Assessment Category	%
Execution correctness	40%
Structure (e.g., modularization, information hiding, etc.)	30%
Insightful programming (e.g., developing reusable class components, etc.)	10%
Consistent style (e.g., capitalization, indenting, etc.)	10%
Appropriate commenting style	10%

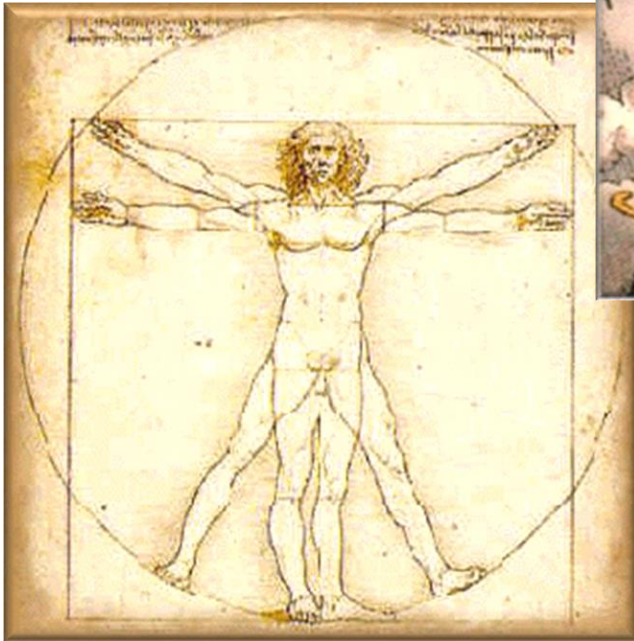


# Summary



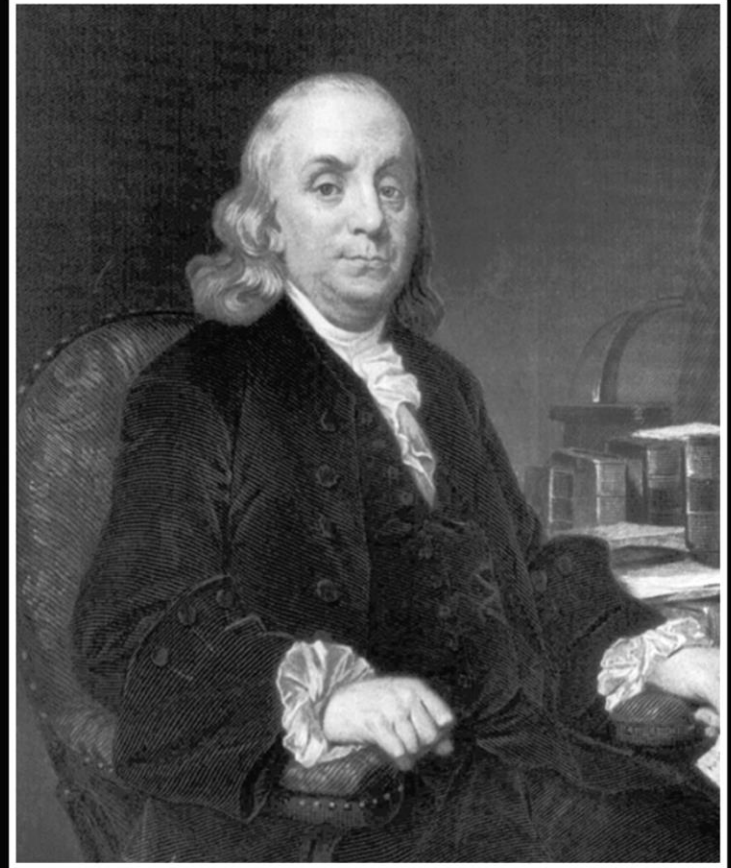
# Summary

- 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

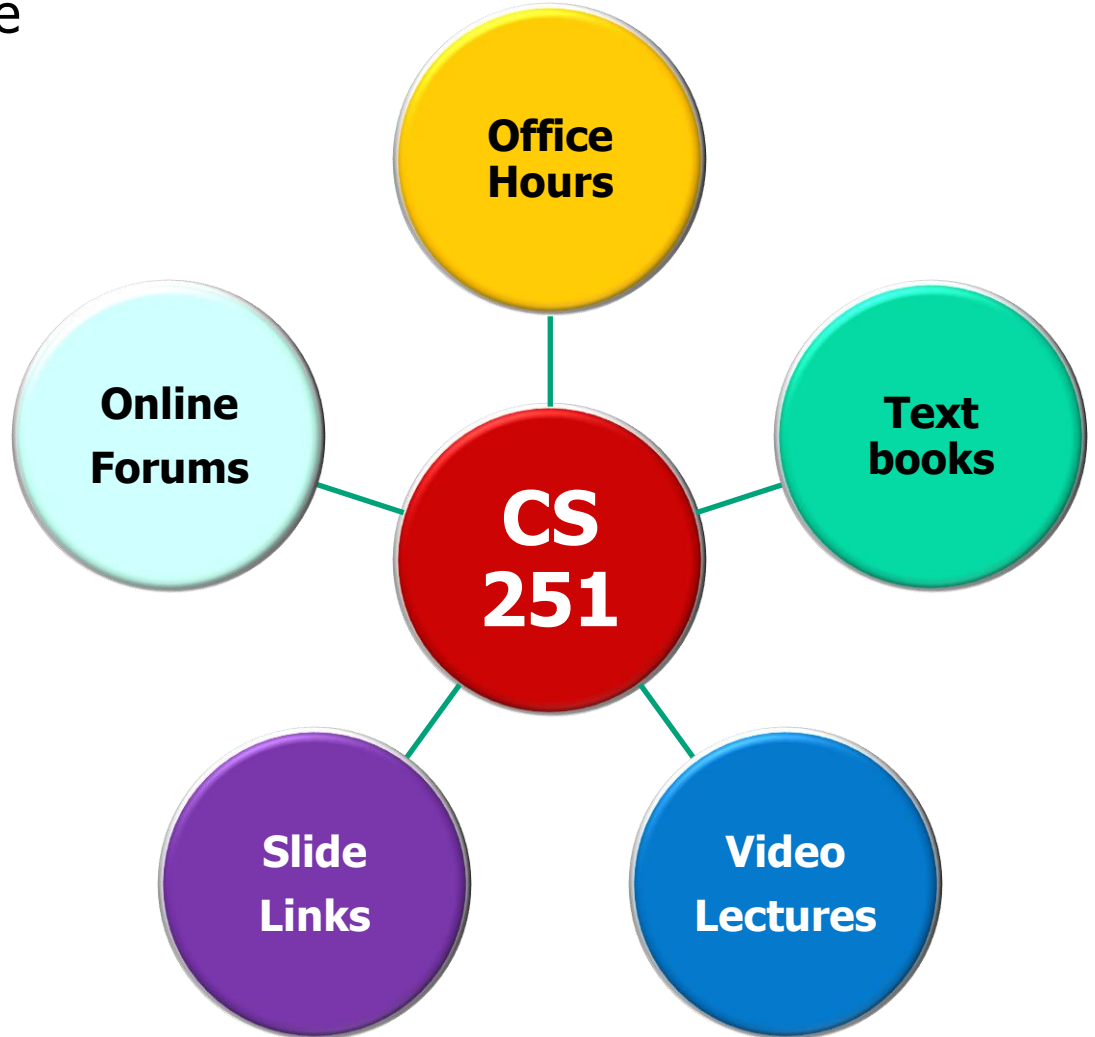


See [piazza.com/vanderbilt/summer2020/cs3251](https://piazza.com/vanderbilt/summer2020/cs3251)

# 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



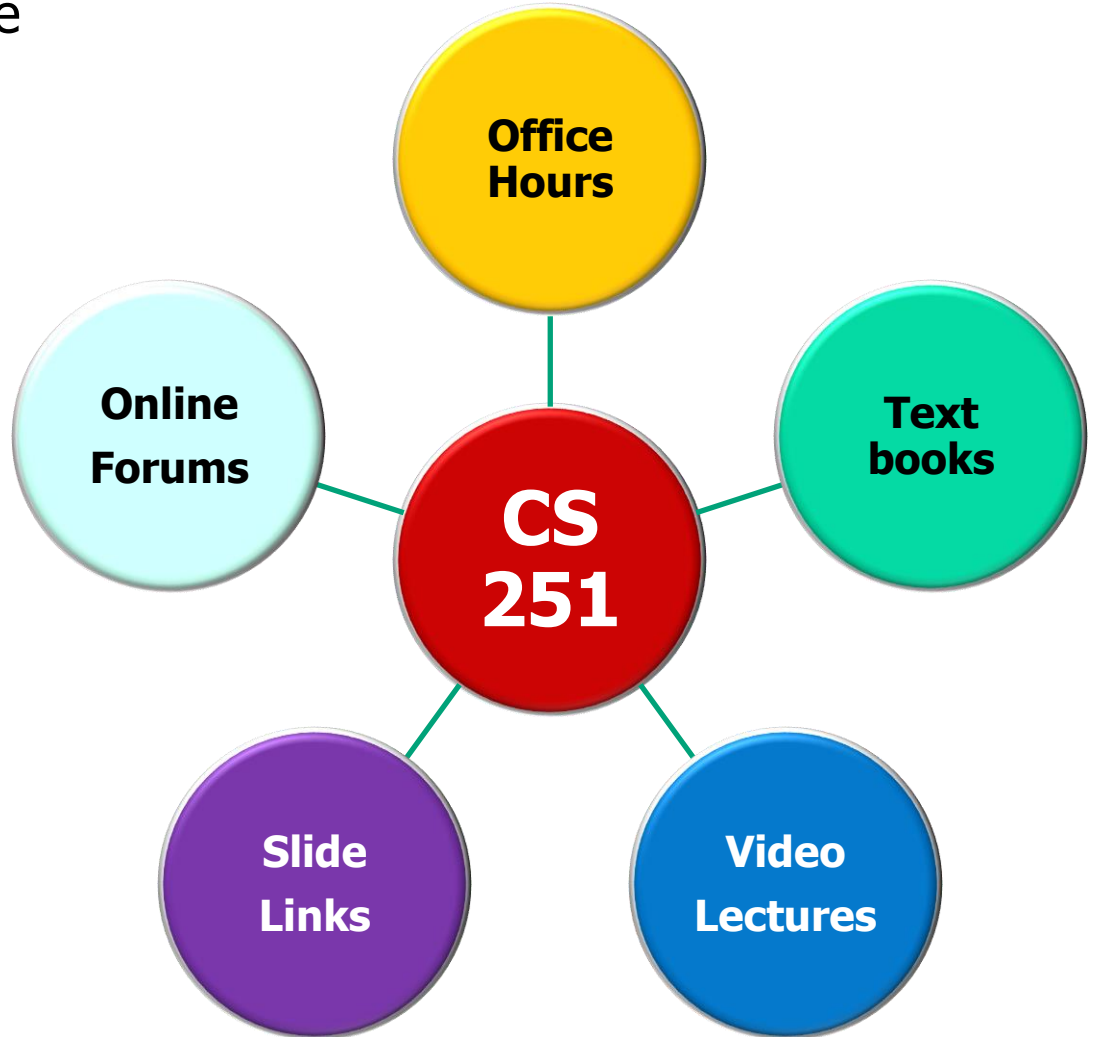
---

See [www.dre.vanderbilt.edu/~schmidt/cs251](http://www.dre.vanderbilt.edu/~schmidt/cs251)

# 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



---

Please resist the urge to email me directly!



# 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
- There are abundant opportunities!



---

See [www.naceweb.org/job-market/compensation/starting-salary-projections-for-top-earning-degrees-level/](http://www.naceweb.org/job-market/compensation/starting-salary-projections-for-top-earning-degrees-level/)