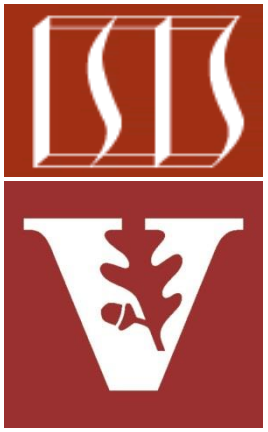
CS *251 Introduction: Course Overview & Logistics

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



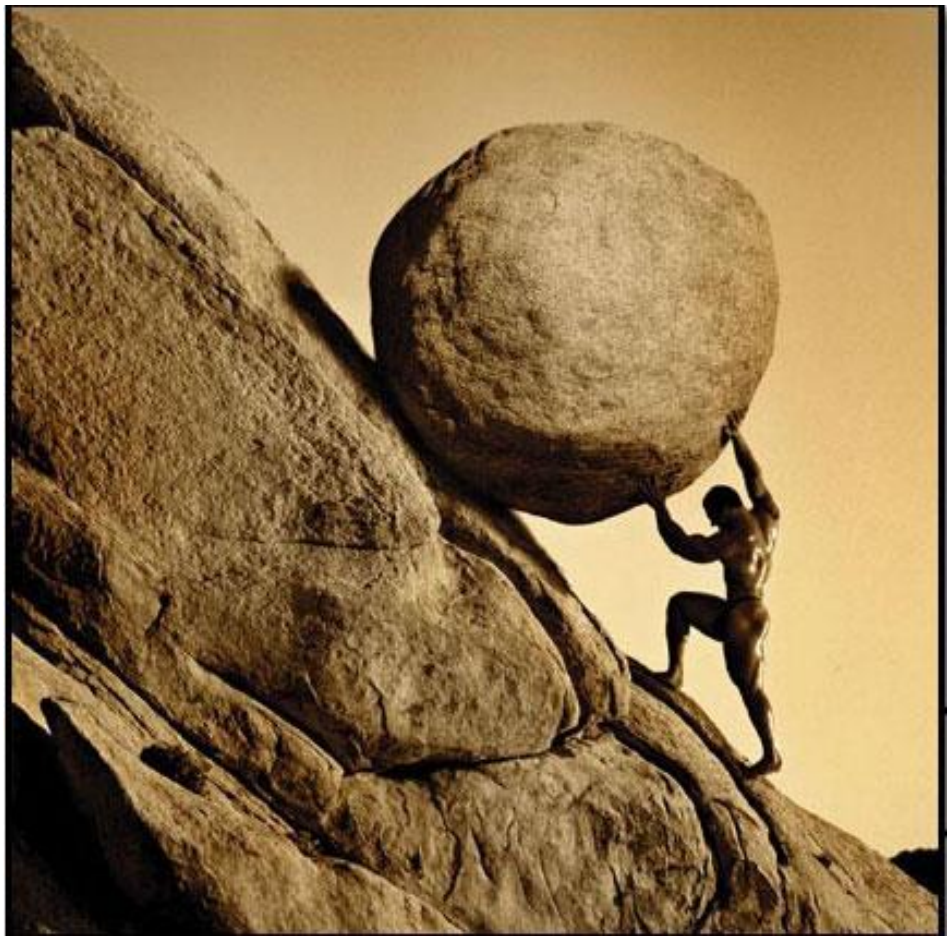
Professor of Computer Science

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

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.

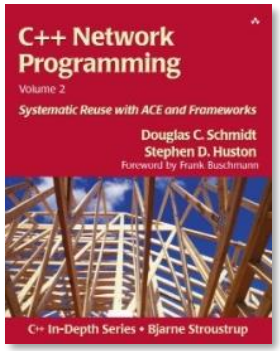
This course involves ***lots*** of hands-on software review, development, & testing

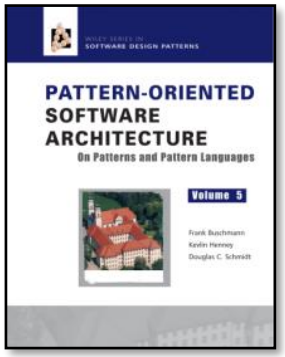
Foundations for this Course Philosophy

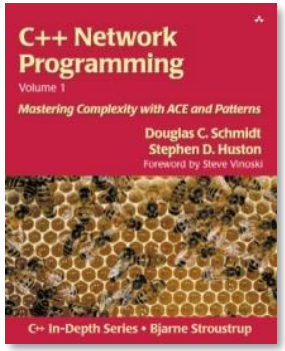
- I've been programming C++ since 1986

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
& en.wikipedia.org/wiki/C++_Report

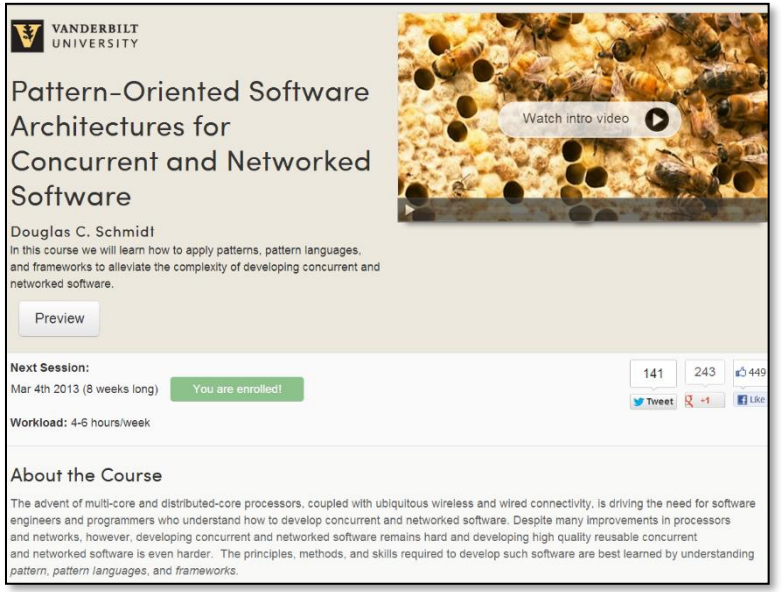
Foundations for this Course Philosophy

- 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

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



See 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

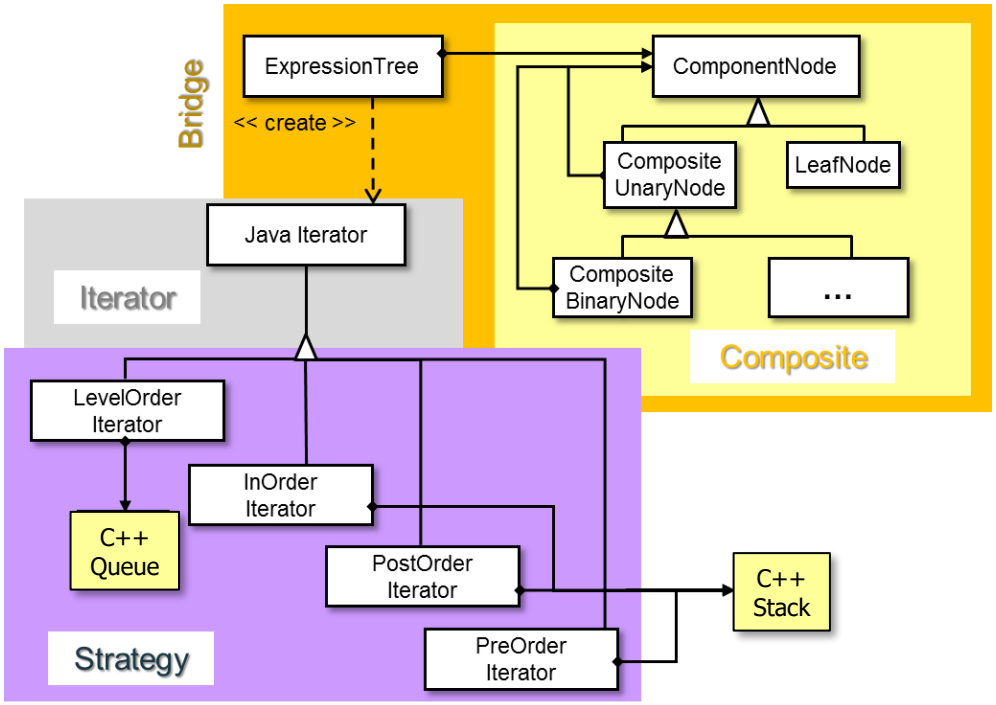
- Topics important to developing & evolving quality C++ software
  - Reuse of patterns & software components
  - Developing, documenting, testing, & applying reusable classes & frameworks

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

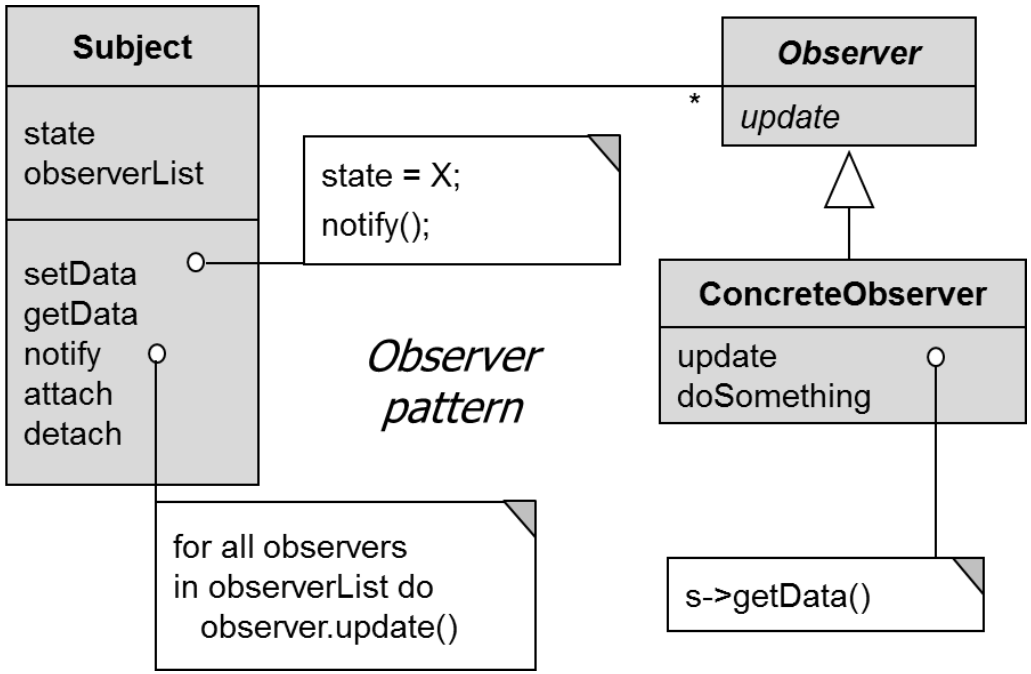
Summary of Course Contents

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

See github.com/douglascraigschmidt/CPlusPlus

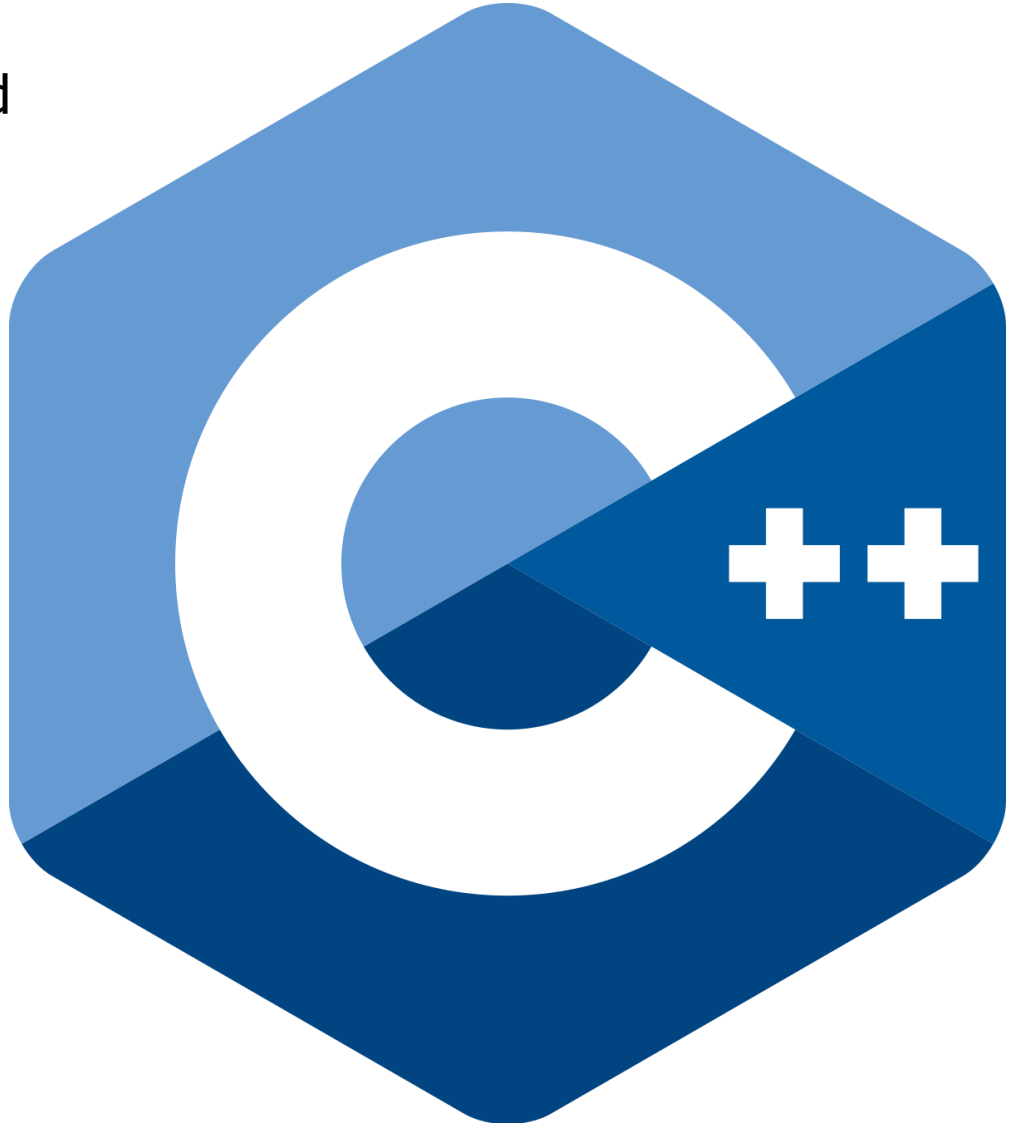
Summary of Course Contents

- 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



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	• Overview of basic object-oriented & generic programming features • Overview of the Clion IDE
Advanced C++ Programming	• C++11 & C++14 features
Standard Template Library (STL)	• Containers, iterators, algorithms
Design Patterns	• Expression tree case study

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

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

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

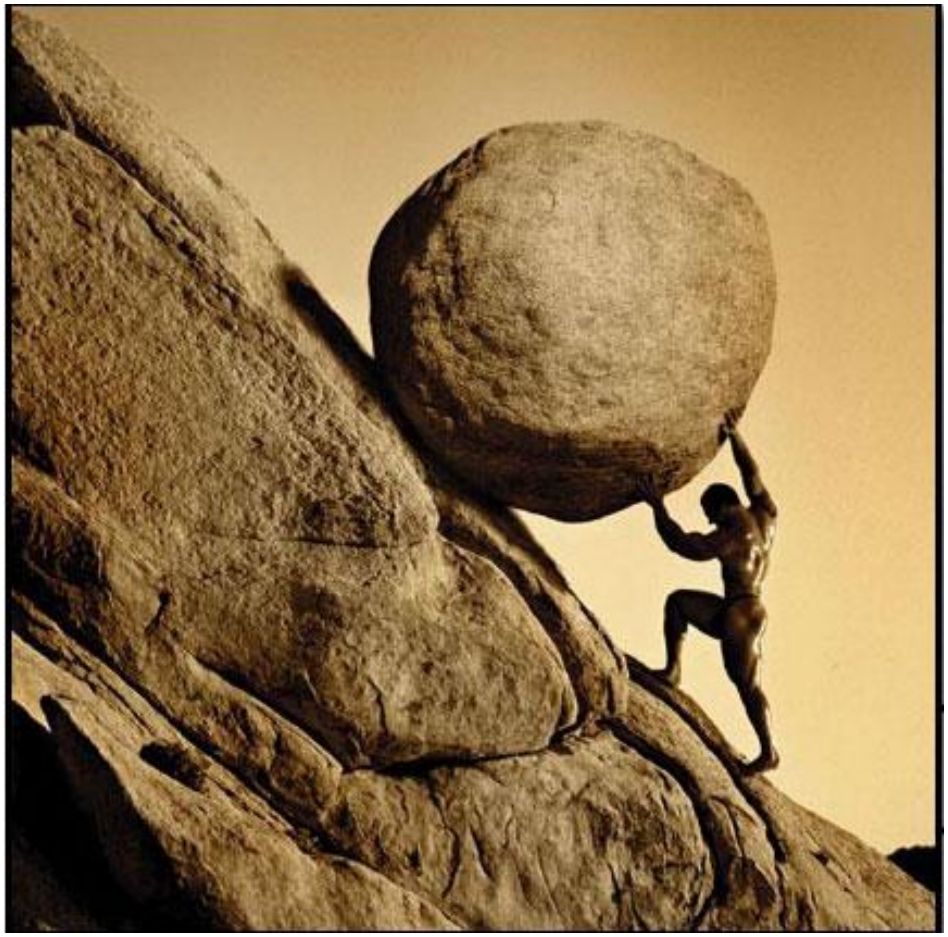
- This course has four main sections
- There will be a weekly quiz on material covered in the lectures
- There will be a cumulative final exam that covers all the lectures

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

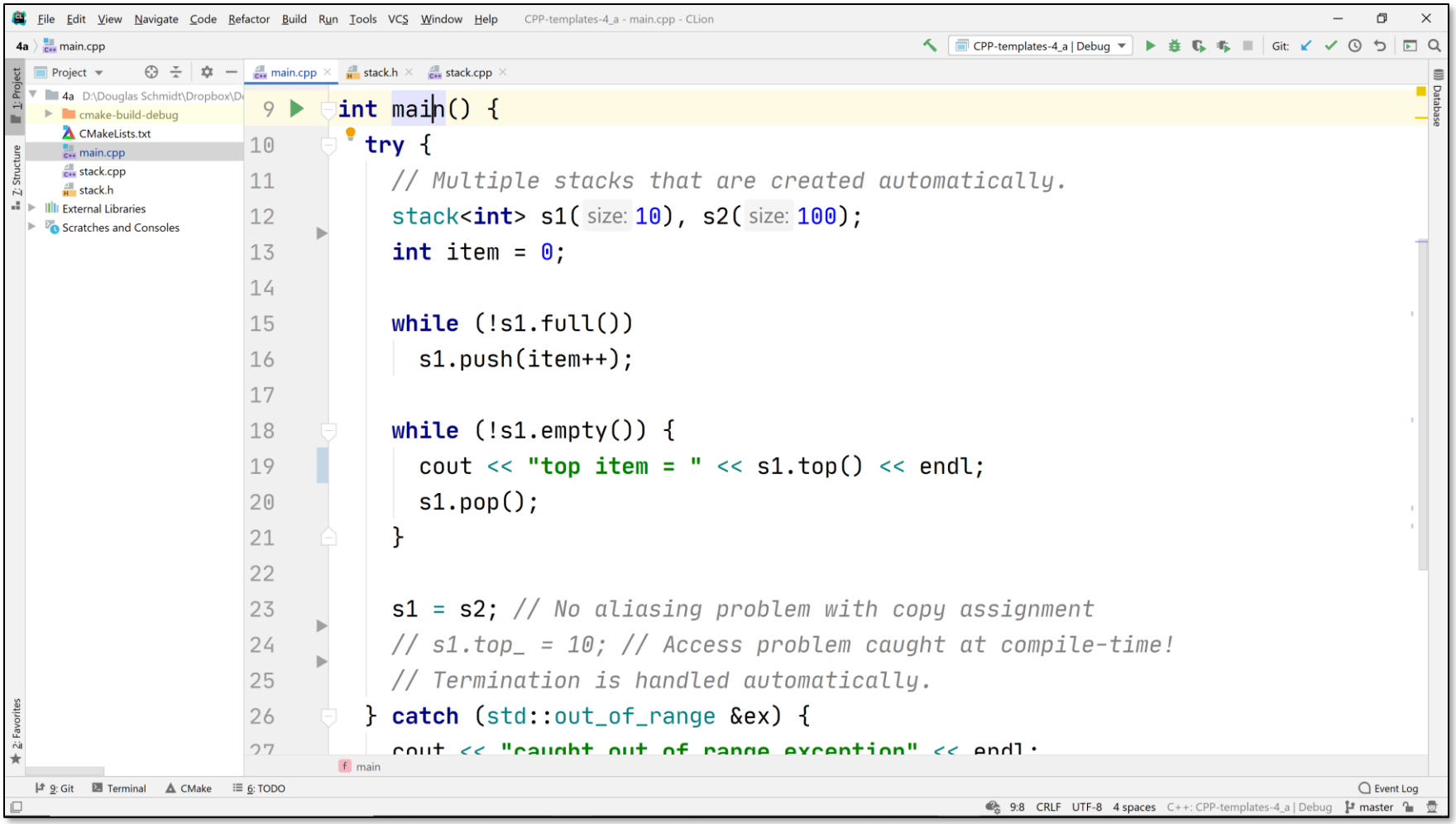
  * Relative weighting may change



Overview of the Assignments & Assessments

Overview of Assignments & Assessments

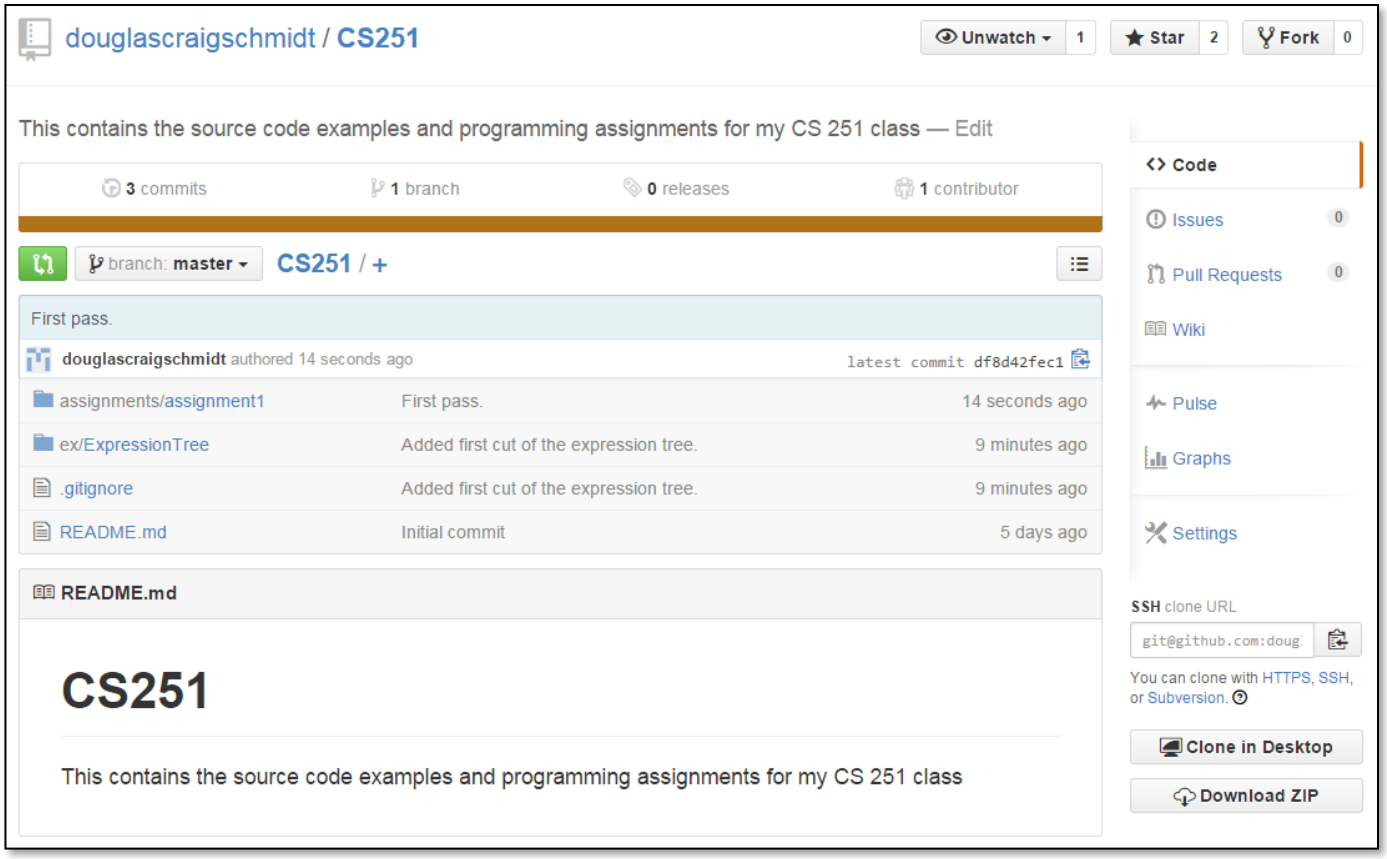
- Programming assignments should be written in C++ using Clion 2020.1



See github.com/douglascraigschmidt/CS251/wiki/Installing-Software

Overview of Assignments & Assessments

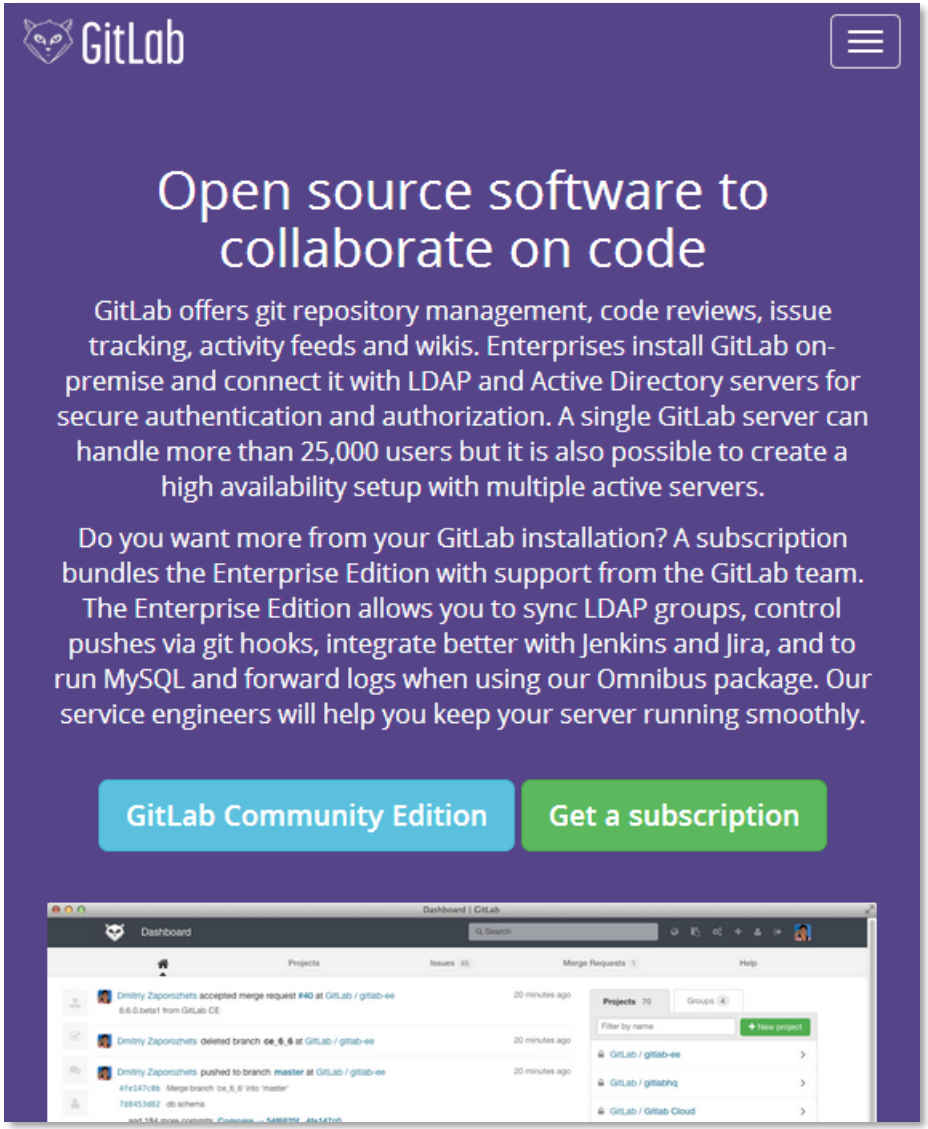
- Programming assignments should be written in C++ using Clion 2020.1
- All source code for assignments & examples available at GitHub



Go to GitHub at github.com/douglascraigschmidt/CS251

Overview of Assignments & Assessments

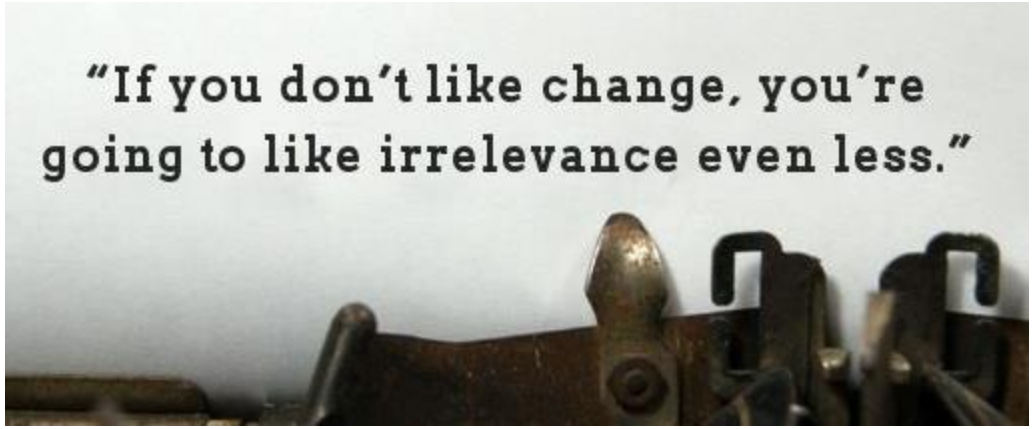
- Programming assignments should be written in C++ using Clion 2020.1
- All source code for assignments & examples available at GitHub
  - You will need to learn how to use GitLab et al.



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



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/douglascraigschmidt/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/douglascraigschmidt/CS251/tree/master/assignments

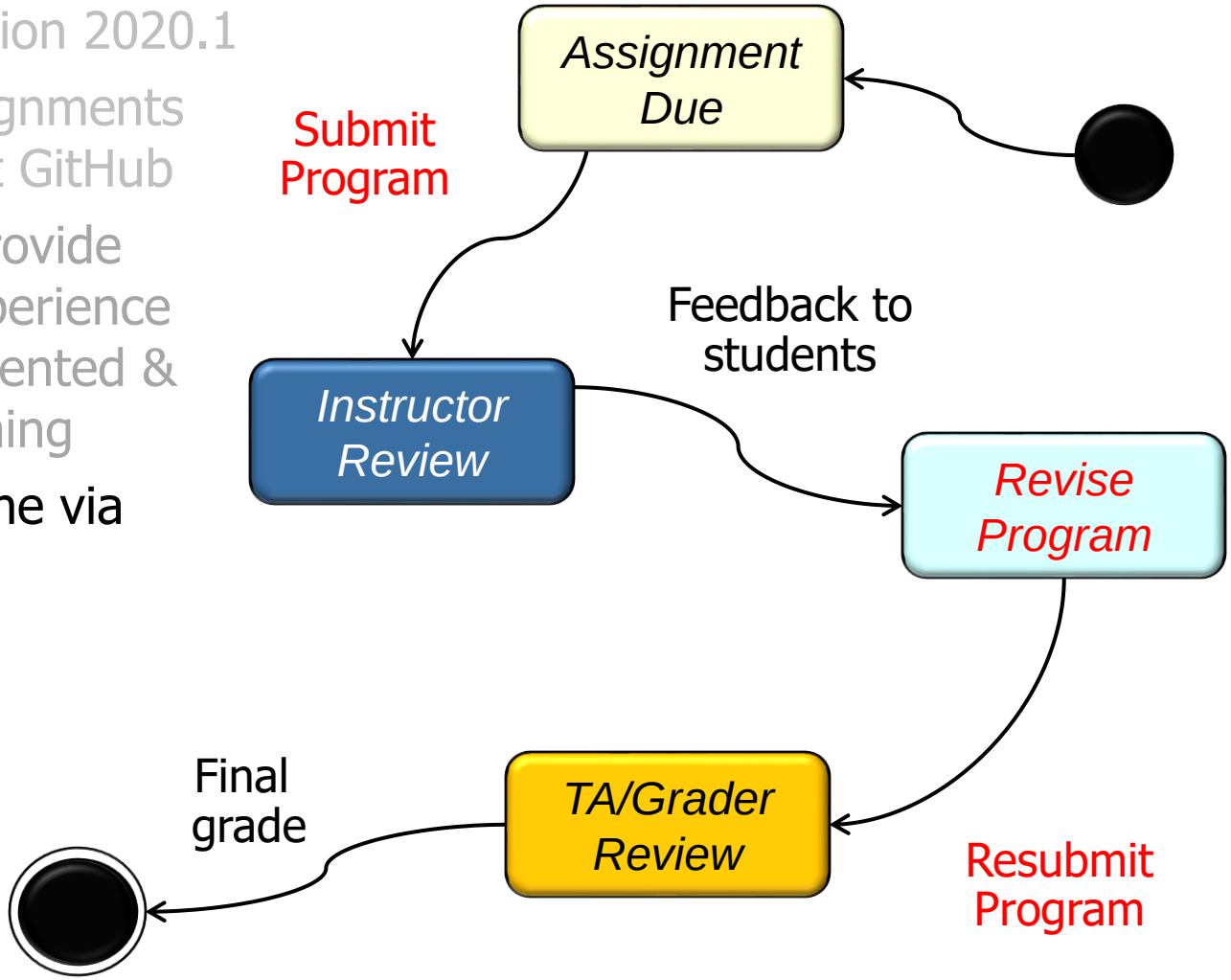
Overview of Assignments & Assessments

- 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

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



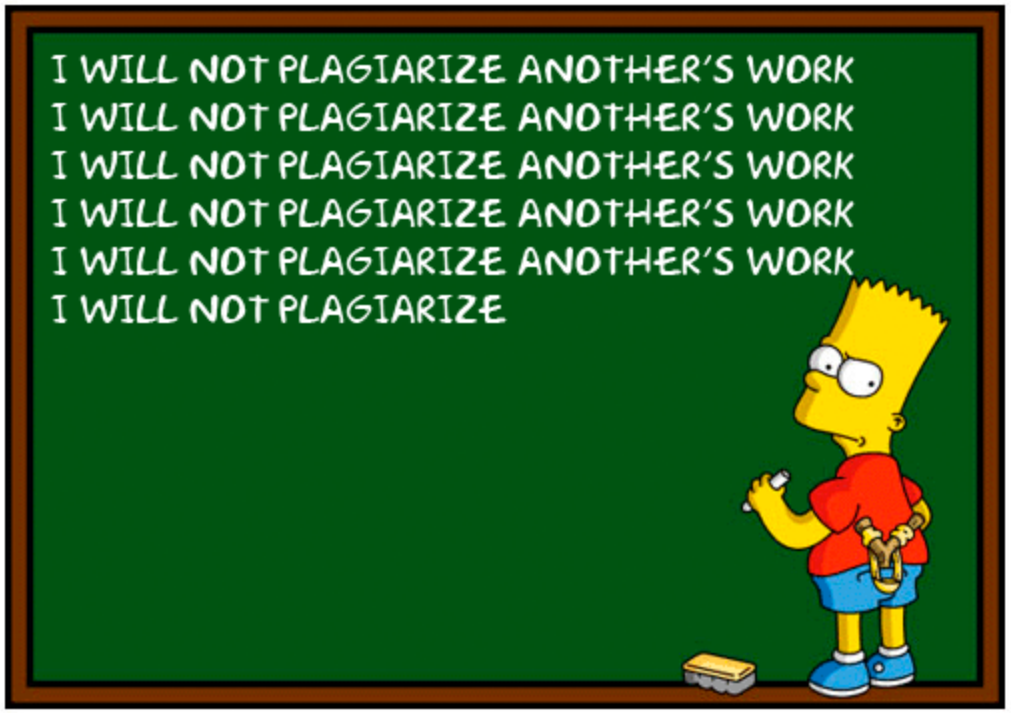
Overview of Assignments & Assessments

- 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

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



www.vanderbilt.edu/student_handbook/the-honor-system#statement-of-the-honor-code

Overview of Assignments & Assessments

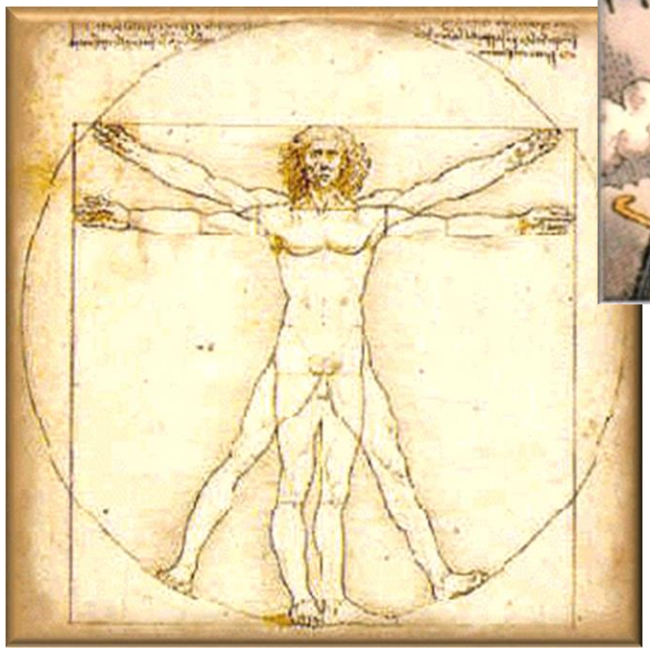
- Programming assignments should be written in C++ using Clion 2020.1
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with pattern-/object-oriented & generic C++ programming
- Assessments will be done via reviews by course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

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

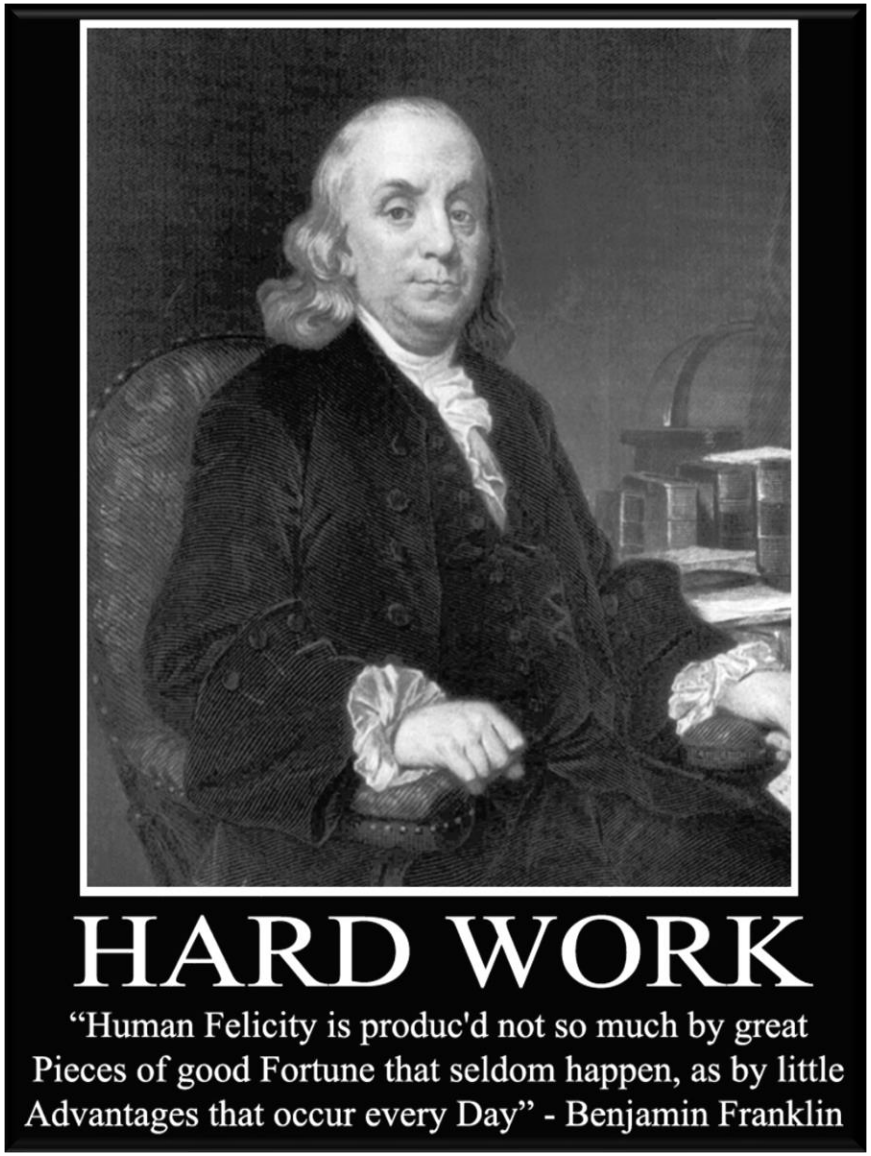
Summary

- 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



Summary

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

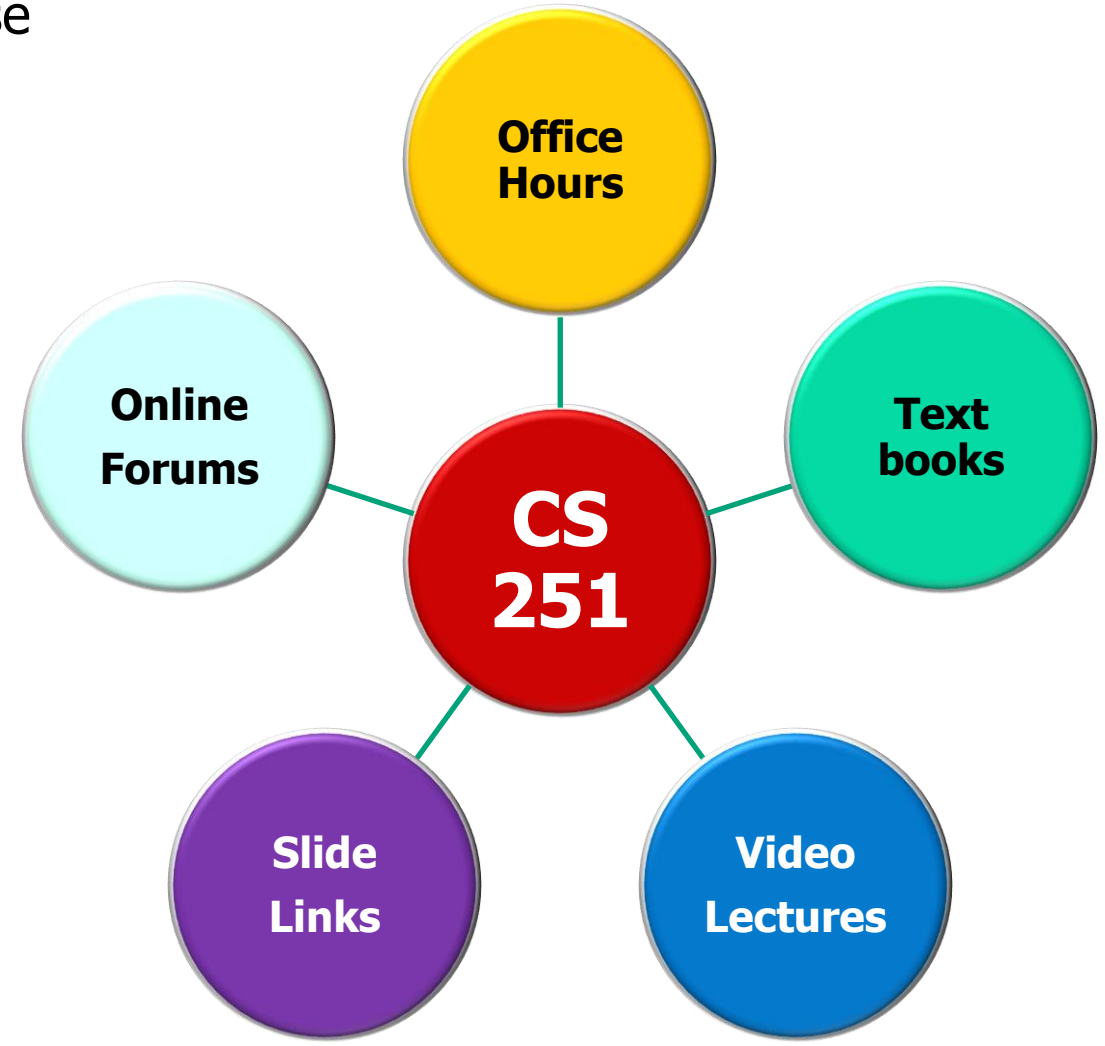
Summary

- 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

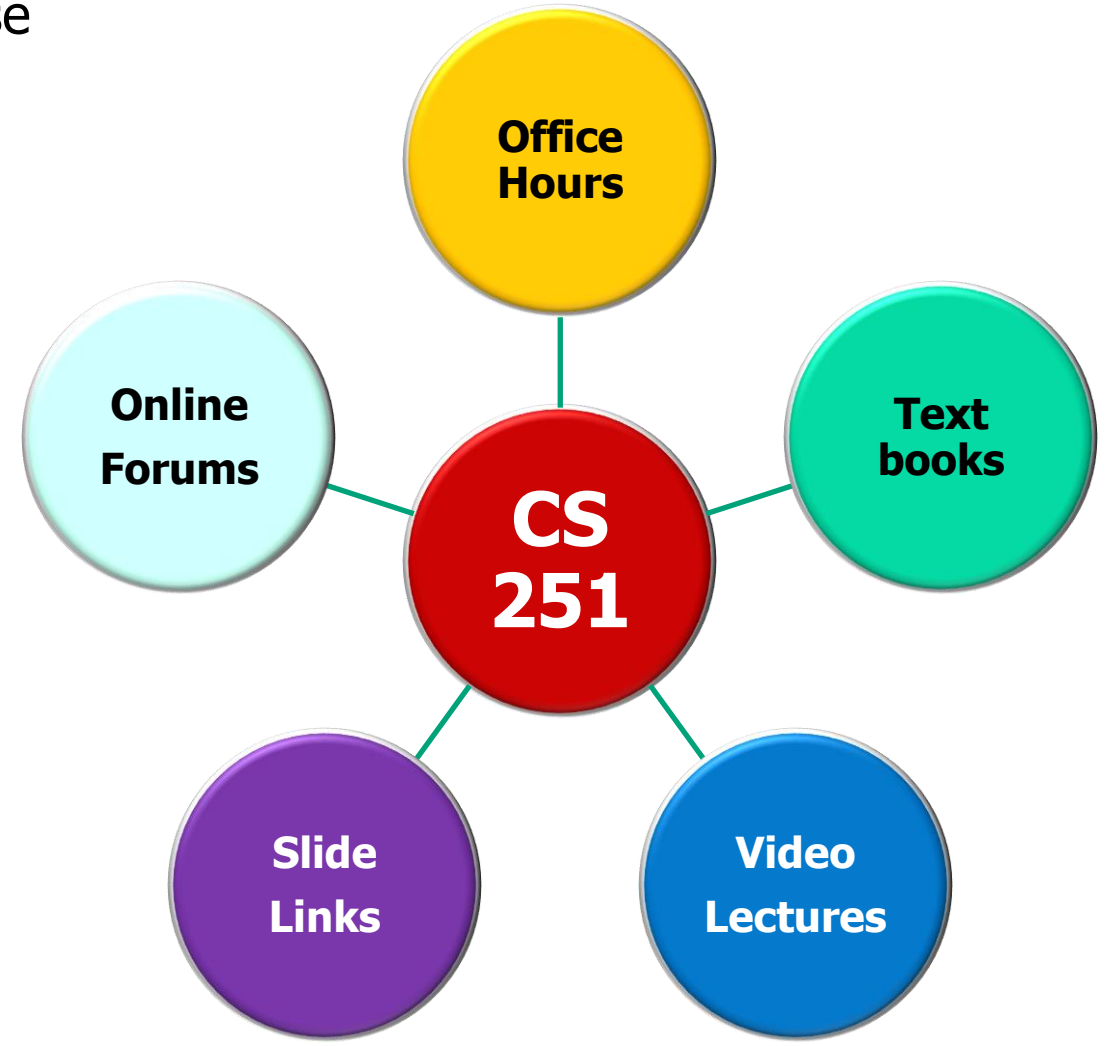
- You will get out of this course what you put into it
  - Be prepared to work hard
  - Do *not* miss deadlines…
  - Participate
  - Avail yourself of available resources



See www.dre.vanderbilt.edu/~schmidt/cs251

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/