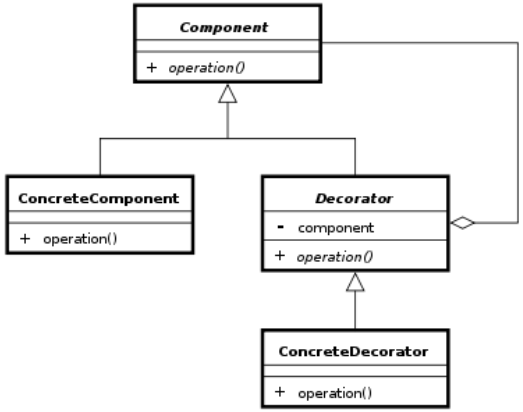
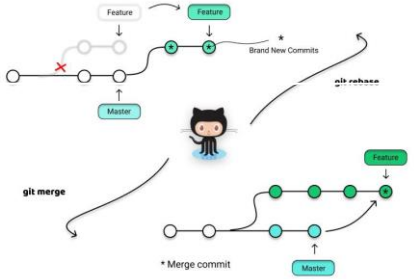
CSci 3081W: Program Design and Development

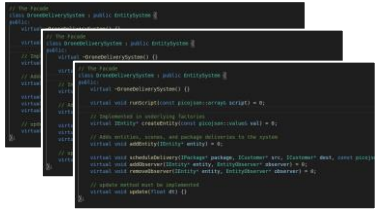
Lecture 01 - Introduction and Logistics

Welcome!

Program Design & Development







C++ Course

CSci 3081 Instructor Crew

Instructors

Shonal Gangopadhyay

Email: gango010@umn.edu

Ace Kaung

Email: kaung006@umn.edu

Teaching Assistants

Primary TAs:

- Lab Section 2 - Corey Knutson
- Lab Section 3 - Zahara Spilka
- Lab Section 4 - Corey Knutson
- Lab Section 5 - Shivam Bhandari
- Lab Section 6 - Shivam Bhandari
- Lab Section 11 - Euna Khan
- Lab Section 12 - Euna Khan
- Lab Section 13 - Tanisha Shrotriya
- Lab Section 14 - Tanisha Shrotriya
- Lab Section 15 - Ying Lu
- Lab Section 16 - Ying Lu

- **Undergraduate TA**
  - Brett Schumacher, Anthony Narlock, Reid Lambert, Sriram Nutulapati, Abdelrahman Mokbel, Fletcher Gornick, Luke Wiskus, Ritik Mishra, Jinming Chen, Drew Young, Daniel Kong, Dominic Deiman, Melissa Tan, Zirui Chen, Ashwin Wariar, Ryan Exner

Staff Email: csci3081s22@umn.edu

Our course is about building the “BIG” project.

Autonomous Vehicle Transportation (e.g. Uber++)

Show the empty demo!

Task - Project Planning: How would you build this system?

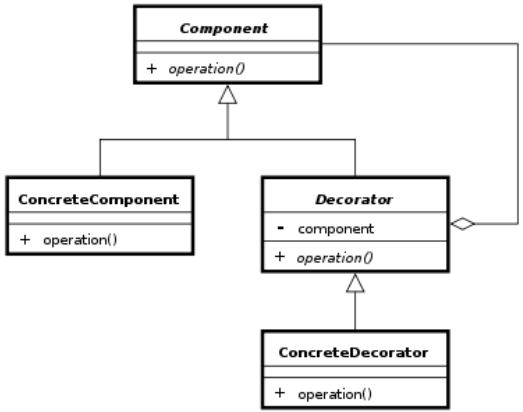
Autonomous Vehicle Transportation (e.g. Uber++)

Draw a diagram of your system on paper. (Work together)

- What components would it have?
- How would different parts of your system communicate?
- What technologies would you use?
- How would you extend your system?

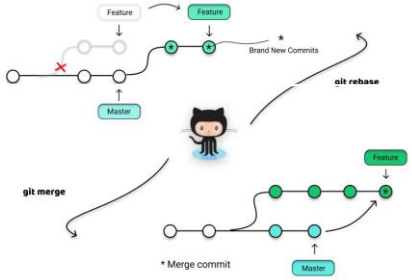
This is a *“practical”* **Design Class**.

Design



*How can we best **design** a flexible system?*

Development



*How can we best **build** a flexible system?*

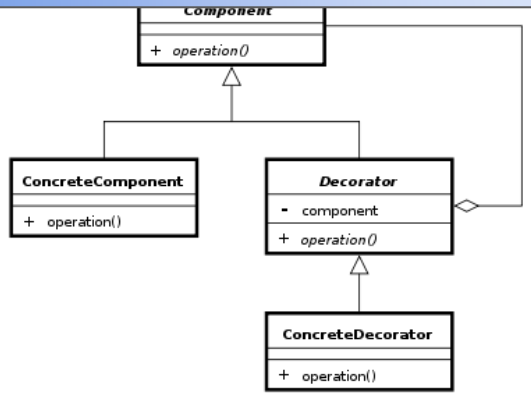
This is a *"practical"* **Design Class**.

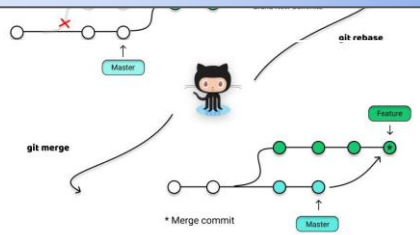
Design

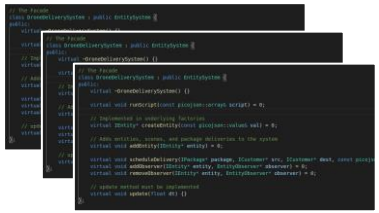
Development

6 Weeks

8 Weeks







*How can we best **design** a flexible system?*

*How can we best **build** a flexible system?*

This is a **writing** class: 3081**W**

Design

*How can we best **design** a flexible system?*

Development

*How can we best **build** a flexible system?*

Design Goals: How can we build a system that is flexible and fast?

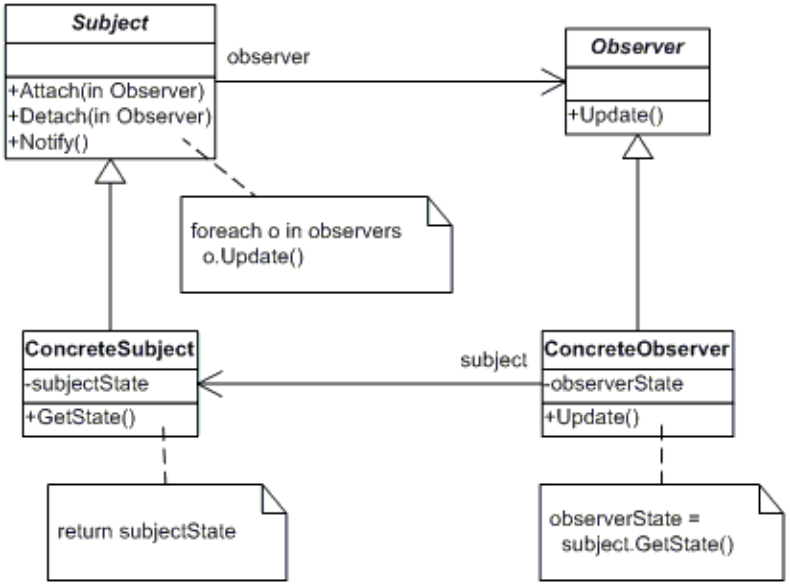
Flexibility ⟷ Efficiency

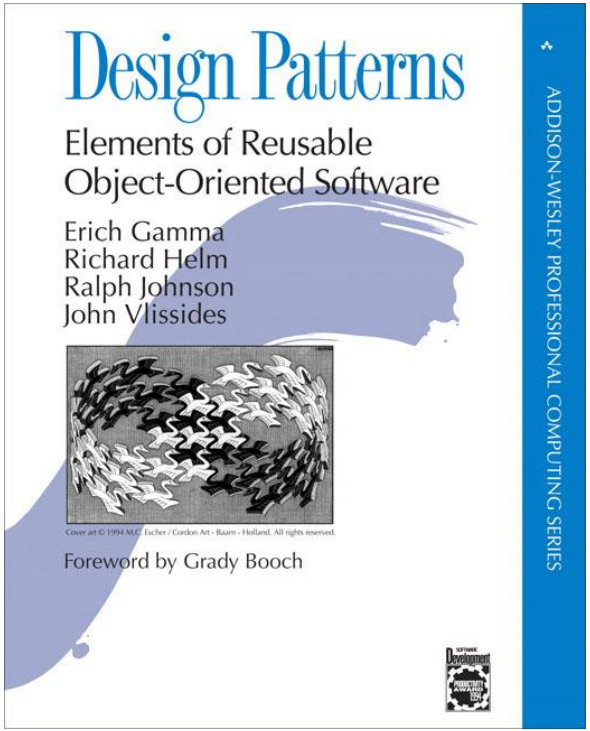
VS

Code Reuse
Design Patterns
Extensibility

Fast Execution
Memory Efficient
Hardware Optimization

For example: We will learn Object Oriented Design





Development Goals: How do we build such a system?

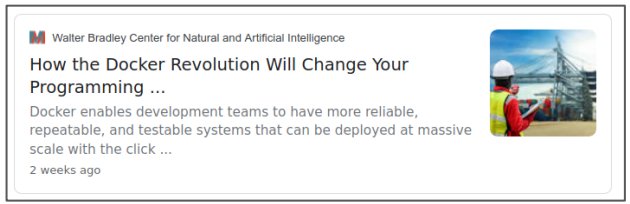
Collaboration

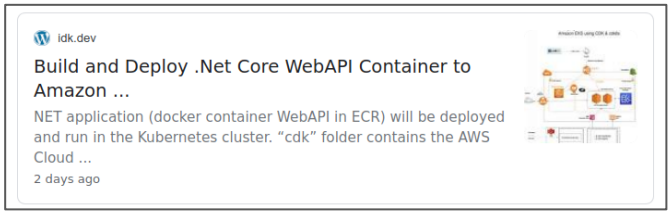
Process

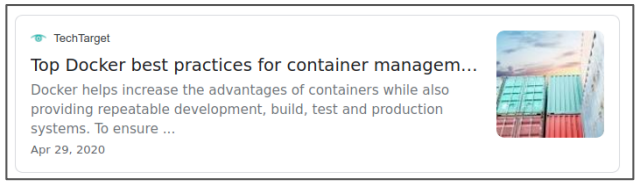
Automation

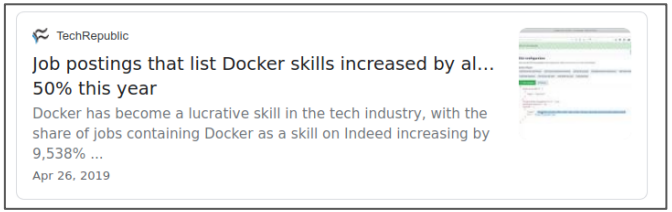
Robustness

For example: We will learn and use Container Architectures.

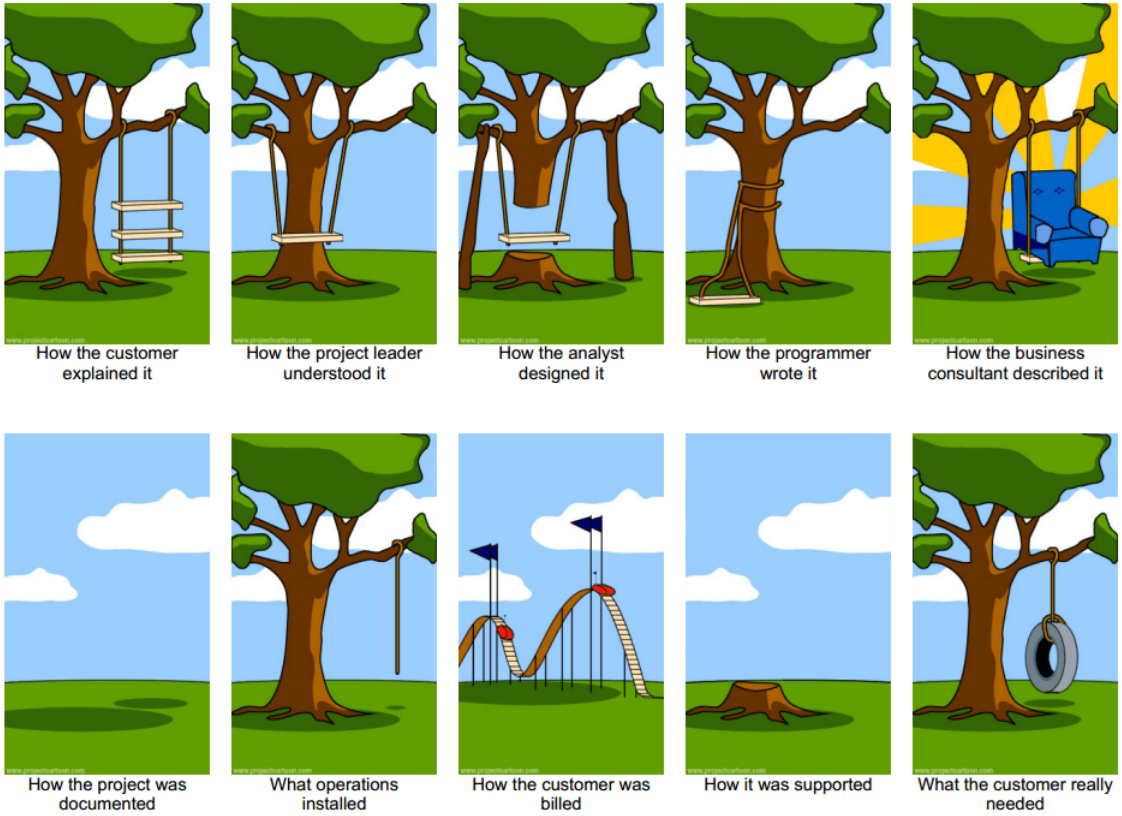








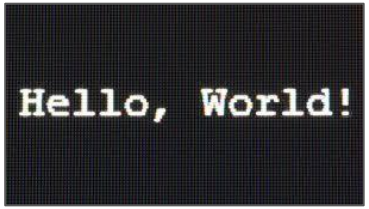
We will introduce some **software engineering** topics.



Course Objectives

- Build a large project from the ground up using best programming practices, good organization, and state of the art technologies.
- Become proficient in C++, memory management, and object oriented programming.
- Effectively follow development processes including version control, code reviews, project management, documentation, and software testing.
- Appropriately use well known design principles and patterns to solve software development problems to increase flexibility and efficiency of a program.
- Describe software design decisions logically using valid references, figures, UML, results, and discussion.
- Confidently argue for a design by comparing it with accepted design principles and alternative designs.
- Propose a useful extension to an existing system. In the process, function effectively as a team member on a project team, with effectiveness being determined by instructor observations, peer ratings, and self-assessment.

Roadmap for Today



Introduction

Course Logistics

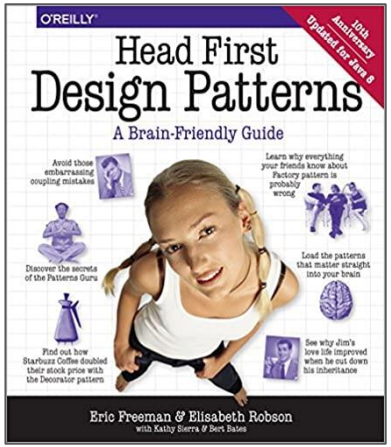
Project Discussion

Please read the **Syllabus** carefully.

Section 01 - https://canvas.umn.edu/courses/333083/pages/course-schedule

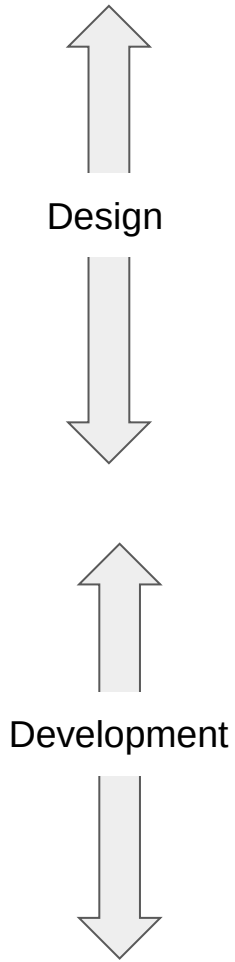
Section 10 - https://canvas.umn.edu/courses/333057/pages/course-schedule

Textbooks



- **zyBook - CSCI 3081W: Program Design and Development C++** (https://learn.zybooks.com/zybook/UMNCSCI3081WFall2022) REQUIRED
- **Head First Design Patterns** (available through the UMN libraryLinks to an external site., at the campus bookstore, and from Amazon or Barnes and Noble). OPTIONAL

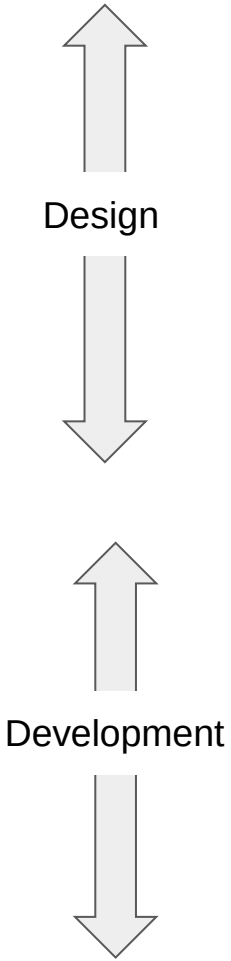
Course Overview



Week	Monday	Reading	Lecture Topics				Lab (Friday)	Homework
			Tuesday		Thursday			
1	Sept 5	Syllabus & Zybooks 1	Intro	Video	Git and Version Control	Video	Lab 01 - Git	
2	Sept 12	Zybooks 2	Workshop 1 - C++	Video	Classes/Structs	Video	Lab 02 - Build Automation and Classes	Homework 1
3	Sept 19	Zybooks 3	Memory Management	Video	Quiz 1		Lab 03 - C++ Memory and Debugging	
4	Sept 26		Workshop 2 - Polymorphism, Abstract, Inheritance	Video	Polymorphism/ Abstraction/ Inheritance	Video	Lab 04 - Code Styling	
5	Oct 3		Polymorphism/ Abstraction/ Inheritance	Video	Unit testing Gtest framework	Video	Lab 05 - Google Testing	Homework 2
6	Oct 10		Workshop 3 - UML	Video	Quiz 2		Lab 06 - Project base code	
7	Oct 17		Façade Factory	Video	Factory	Video	Lab 07 - Factory Design Pattern	
8	Oct 24		Workshop 4 - Factories	Video	Strategy	Video	Lab 08 - Strategy Design Pattern	Homework 3
9	Oct 31		Observer	Video	Decorator	Video	Lab 09 - Decorator Design Pattern	
10	Nov 7		Workshop 5 - Docker	Video	Quiz 3		Lab 10 - Doxygen	
11	Nov 14		Development Processes	Video	Development Processes	Video	Lab 11 - HW4/Project Brainstorming	Homework 4
12	Nov 21		Workshop 6 - Agile/Scrum	Video	**Thanksgiving**	Video	**Thanksgiving**	
13	Nov 28		Development Processes	Video	Quiz 4		Lab 13 - Project	
14	Dec 5			Video		Video	Lab 14 - Project	
15	Dec 12			Video		Video	Lab 15 - Presentation	

Course Overview

- 5% zyBooks
- 10% Workshops
- 50% Homework
- 20% Quizzes
- 15% Lab



Week	Monday	Reading	Lecture Topics				Lab (Friday)	Homework
			Tuesday		Thursday			
1	Sept 5	Syllabus & Zybooks 1	Intro	Video	Git and Version Control	Video	Lab 01 - Git	
2	Sept 12	Zybooks 2	Workshop 1 - C++	Video	Classes/Structs	Video	Lab 02 - Build Automation and Classes	Homework 1
3	Sept 19	Zybooks 3	Memory Management	Video	Quiz 1		Lab 03 - C++ Memory and Debugging	
4	Sept 26		Workshop 2 - Polymorphism, Abstract, Inheritance	Video	Polymorphism/ Abstraction/ Inheritance	Video	Lab 04 - Code Styling	
5	Oct 3		Polymorphism/ Abstraction/ Inheritance	Video	Unit testing Gtest framework	Video	Lab 05 - Google Testing	Homework 2
6	Oct 10		Workshop 3 - UML	Video	Quiz 2		Lab 06 - Project base code	
7	Oct 17		Façade Factory	Video	Factory	Video	Lab 07 - Factory Design Pattern	
8	Oct 24		Workshop 4 - Factories	Video	Strategy	Video	Lab 08 - Strategy Design Pattern	Homework 3
9	Oct 31		Observer	Video	Decorator	Video	Lab 09 - Decorator Design Pattern	
10	Nov 7		Workshop 5 - Docker	Video	Quiz 3		Lab 10 - Doxygen	
11	Nov 14		Development Processes	Video	Development Processes	Video	Lab 11 - HW4/Project Brainstorming	Homework 4
12	Nov 21		Workshop 6 - Agile/Scrum	Video	**Thanksgiving**	Video	**Thanksgiving**	
13	Nov 28		Development Processes	Video	Quiz 4		Lab 13 - Project	
14	Dec 5			Video		Video	Lab 14 - Project	
15	Dec 12			Video		Video	Lab 15 - Presentation	

Course Overview

5% • **zyBooks**

25% • Labs / Workshops

50% • Homework

20% • Quizzes

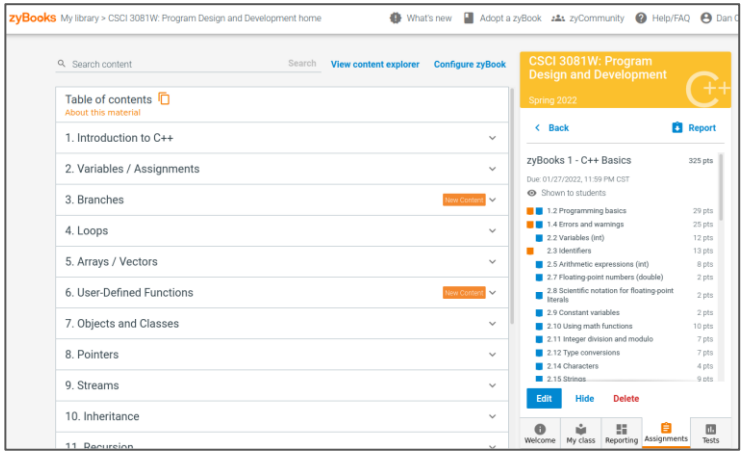
Purpose: Learn C++ Asynchronously

---

1. Sign in or create an account at learn.zybooks.com (Links to an external site.)
2. Enter zyBooks code: UMNCSCI3081WFall2022
3. Subscribe

---

https://learn.zybooks.com/zybook/UMNCSCI3081WFall2022



Allows us to focus more on design and development.

Course Overview

5%	•	zyBooks
25%	•	**Labs / Workshops**
50%	•	Homework
20%	•	Quizzes

Labs: We will build the transportation system through the labs.

You will gain hands on experience working with:

- Javascript / HTML / CSS
- Web Sockets
- Git
- Object Oriented Programming
- Memory Management
- Design Patterns
- Testing
- Documentation
- More…

Course Overview

5% • zyBooks

25% • **Labs / Workshops**

50% • Homework

20% • Quizzes

Note: Labs will often be automatically graded. However, it is impossible to write tests that will work for every possible situation. Therefore, if the automated test fails, you may request manual grading.

Labs: We will build the transportation system through the labs.

You will gain hands on experience working with:

- Javascript / HTML / CSS
- Web Sockets
- Oriented Programming
- Management
- Patterns
- entation
- More…

Course Overview

5% • zyBooks

25% • **Labs / Workshops**

50% • Homework

20% • Quizzes

This class is organized by Lab Section!

- Lab Section 2 - Corey Knutson
- Lab Section 3 - Zahara Spilka
- Lab Section 4 - Corey Knutson
- Lab Section 5 - Shivam Bhandari
- Lab Section 6 - Shivam Bhandari
- Lab Section 11 - Euna Khan
- Lab Section 12 - Euna Khan
- Lab Section 13 - Tanisha Shrotriya
- Lab Section 14 - Tanisha Shrotriya
- Lab Section 15 - Ying Lu
- Lab Section 16 - Ying Lu

Your Lab section defines you TAs for the semester.

Course Overview

5%	•	zyBooks
25%	•	Labs / **Workshops**
50%	•	Homework
20%	•	Quizzes

Workshops: There will be 6 In-Class coding workshops.

- Students will be given a lecture relevant problem to solve by the end of class.
- Always graded by participation.

You will gain hands on experience working with:

- Object Oriented Design
- Basic C++
- UML
- Simulation
- Agile Process
- Unit Testing
- Design Problems

Course Overview

5% ● zyBooks

25% ● Labs / Workshops

50% ● **Homework**

20% ● Quizzes

Homework: There will be 4 homework assignments.

- More writing intensive.
- Focus on learning objectives beyond project and programming.
- Introduce new technologies or paradigms.
- Often involve students from the class.

Course Overview

5% • zyBooks

25% • Labs / Workshops

50% • **Homework**

20% • Quizzes

Homework: There will be 4 homework assignments.

- More writing intensive.
- Focus on learning objectives beyond project and programming.
- Introduce new technologies or paradigms.
- Often involve students from the class.

Homework 4: The Proposal

- You will work in a team to propose useful functionality beyond the project we developed.
- You and your team will “make progress” towards this goal.
- You will present your proposal to the class as a team.

Course Overview

5%	•	zyBooks
25%	•	Labs / Workshops
50%	•	Homework
20%	•	**Quizzes**

Quizzes: There will be 4 take home quizzes (online Gradescope).

- These will cover topics from lecture, labs, workshops, and homework.
- There is no study guide for these quizzes. Any material covered before a quiz is fair game on a quiz.
- You can start the quiz at any time in 24 hours, and finish by the allotted time.

Office Hours

According to your Lab Section. Each lab is assigned at least one Graduate TA and two Undergraduate TAs. Some sections have more TAs due to number of labs and lab sizes.

- Lab Section 2 - Corey Knutson
- Lab Section 3 - Zahara Spilka
- Lab Section 4 - Corey Knutson
- Lab Section 5 - Shivam Bhandari
- Lab Section 6 - Shivam Bhandari

- Lab Section 11 - Euna Khan
- Lab Section 12 - Euna Khan
- Lab Section 13 - Tanisha Shrotriya
- Lab Section 14 - Tanisha Shrotriya
- Lab Section 15 - Ying Lu
- Lab Section 16 - Ying Lu

Office Hours

According to your Lab Section. Each lab is assigned at least one Graduate TA and two Undergraduate TAs. Some sections have more TAs due to number of labs and lab sizes.

- Lab Section 2 - Corey Knutson
- Lab Section 3 - Zahara Spilka
- Lab Section 4 - Corey Knutson
- Lab Section 5 - Shivam Bhandari
- Lab Section 6 - Shivam Bhandari

- Lab Section 11 - Euna Khan
- Lab Section 12 - Euna Khan
- Lab Section 13 - Tanisha Shrotriya
- Lab Section 14 - Tanisha Shrotriya
- Lab Section 15 - Ying Lu
- Lab Section 16 - Ying Lu

In order for this class to scale, **please only attend office hours of or ask questions to your assigned Lab TAs**.

Late Work Policy

It just doesn't work in this class:

- *We usually talk about solutions in the next class.*
- *The project is iterative and students get behind.*

Please reach out to me if you have any questions, concerns, or **accommodations**.
Exceptions to this rule will only be given for university approved accommodations (illness, DRC, etc…)

Suggestion: Turn in what you have an hopefully you will get partial credit.

Lecture Modality - Currently In-Person

Lectures will be in person unless anything changes to the University Policy.

Lectures will be uploaded by our recording TA for viewing lecture recordings. These will be uploaded within 24 hours after the lecture is given.

Lecture Modality - Currently In-Person

Lectures will be in person unless anything changes to the University Policy.

Lectures will be uploaded by our recording TA for viewing lecture recordings. These will be uploaded within 24 hours after the lecture is given.

Online Exceptions

- We will have Discord Server available for Friday Labs and In-Class Workshops / Exercise.
- Please work with your lab sections's Graduate TA for specific accommodations.
    - This might be detailed in the Canvas Lab Section Syllabus.

Lecture Modality - Prepare and perhaps expect changes.

Our contingency plan for COVID-19 issues:

- Most everything will stay the same (online accommodations for labs, workshops, etc…)
- The class will be flipped:
  - Lectures will be pre-recorded and should be viewed before class.
  - During class we will focus on discussion and in-class activities in a hybrid or zoom / discord mode.
  - These discussions may be led by TAs if I am unavailable.

Topics!

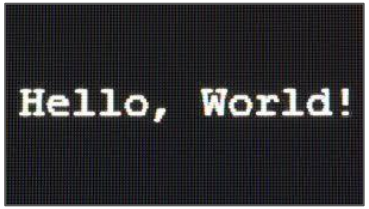
Lab 01
(Due 9/15)
&
zyBooks 1
(Due 9/11)

Week	Monday	Reading	Lecture Topics				Lab (Friday)	Homework
			Tuesday		Thursday			
1	Sept 5	Syllabus & Zybooks 1	Intro	Video	Git and Version Control	Video	Lab 01 - Git	
2	Sept 12	Zybooks 2	Workshop 1 - C++	Video	Classes/Structs	Video	Lab 02 - Build Automation and Classes	Homework 1
3	Sept 19	Zybooks 3	Memory Management	Video	Quiz 1		Lab 03 - C++ Memory and Debugging	
4	Sept 26		Workshop 2 - Polymorphism, Abstract, Inheritance	Video	Polymorphism/ Abstraction/ Inheritance	Video	Lab 04 - Code Styling	
5	Oct 3		Polymorphism/ Abstraction/ Inheritance	Video	Unit testing Gtest framework	Video	Lab 05 - Google Testing	Homework 2
6	Oct 10		Workshop 3 - UML	Video	Quiz 2		Lab 06 - Project base code	
7	Oct 17		Façade Factory	Video	Factory	Video	Lab 07 - Factory Design Pattern	

Topics!

8	Oct 24		Workshop 4 - Factories	Video	Strategy	Video	Lab 08 - Strategy Design Pattern	Homework 3
9	Oct 31		Observer	Video	Decorator	Video	Lab 09 - Decorator Design Pattern	
10	Nov 7		Workshop 5 - Docker	Video	Quiz 3		Lab 10 - Doxygen	
11	Nov 14		Development Processes	Video	Development Processes	Video	Lab 11 - HW4/Project Brainstorming	Homework 4
12	Nov 21		Workshop 6 - Agile/Scrum	Video	**Thanksgiving**	Video	**Thanksgiving**	
13	Nov 28		Development Processes	Video	Quiz 4		Lab 13 - Project	
14	Dec 5			Video		Video	Lab 14 - Project	
15	Dec 12			Video		Video	Lab 15 - Presentation	

Roadmap for Today



Introduction

Course Logistics

Project Discussion

Additional Slides

Why C++?

- Efficient programming is about managing resources.
- Object Oriented practices help us design flexible systems.
- C++ is both of these!

	C	Java	Python	C++
Efficient: Memory Management	✓	✗	✗	✓
Flexible: Object Oriented	✗	✓	✓	✓

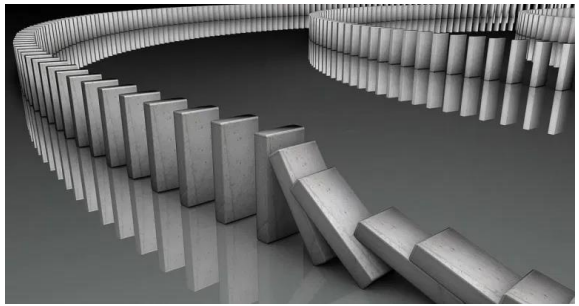
"I wish someone would have told me about memory when I first started programming."

Other really great reasons to use C++.

Computer Graphics

Library Support

System and Device Programming



Standard Template Library

Perhaps the best reason to learn C++.

Once you learn C++, you can pick up nearly any other computer language and understand how it works.