

Meeting Times

- Tuesday, Thursday 12:15 - 13:30 in Reid 103. (August 26 - December 6, 2019)

Textbooks

- *Head First Design Patterns*, by Freeman and Freeman & O'Reilly.
- *A pragmatic introduction to UML*, by Miles and Hamilton.
- *Other resources as communicated by instructor.*

Software and Tools

- You may use any development environment you choose, e.g., Eclipse, NetBeans, etc. As far as computer languages, **only** C++ or Java are acceptable to do assignments. We will also learn and use UML. You are free to use any tool that facilitates generation of UML diagrams as long as you can email them to me such that I can see them (i.e. pdf or any image type). Options include commercial tools (Altova, Rational), OpenSource, Visio, ArgoUML, plantUML, etc. It is your responsibility to experiment with these tools and pick one that suits your needs. Hand drawn UML diagrams **will not** be accepted.

Grading

- Homework: 30%
- Mid Term : 35%
- Final : 35%

Grading Policy

- **Important:** You must get at least 40/70 in your exams (combined) to pass the course. Anything less than 40 is an automatic failure, regardless of your homework score.
- The overall grade for the course will be curved if you meet the 40 point minimum in your exams.
- Homework and exams will be graded on a straight scale. No curve.
- If you come to class, participate, read the assigned material, and complete your homework you should do fine in the course.
- Any cheating will be dealt with severely. You will receive an automatic 'F' and dismissal from the course. I monitor prior year's solutions, Stack Overflow, etc.
- Use of material that is not yours must be cited.

Instructor and TA Information

- Dr. Clem Izurieta NAH 253D. 994-3720. Office hours are Tuesdays and Thursdays 9:00-10:00 or by appointment.
- Email : clemente.izurieta@montana.edu OR via D2L.
- TA: Derek Reimanis, derek.reimanis@msu.montana.edu

Policy

- You are responsible for any announcements made in class.
- **No late submissions are allowed** unless an emergency occurs. **This is strictly enforced.** D2L will not accept late assignments. Please do not challenge this policy.
- Most homework (unless otherwise specified) will be done in teams of two. Once you select a partner you will keep that partner for the remainder of the semester. Choose carefully.
- Academic DISHONESTY results in an **automatic F** for you and the person you copy from. This should never be a problem, however instances of cheating do occur (i.e. StackOverflow). There are web sites in the Internet that provide answers to high bidders. I do monitor these sites. If you do not know how to solve a problem, come see me!
- You are expected to attend class. A lack of attendance will affect you when I curve final grades.
- Absolutely **no texting or using cell** phones in class. If you must use your mobile/cell device you must leave the classroom.

Course Outcomes

- Understand project lifecycles and management basics.
- Learn UML.
- Understand design patterns and be able to identify when to use them.
- Understand the basics of formal methods and testing.
- Have a broad sense of the topics involved in Software Engineering.
- Understand basic measurements and metrics used in Software Engineering.
- We will cover some special topics like Technical Debt, DevOps, etc.

Content

The table below shows the various topics we will cover in this course. In the Reading column I will place expected reading you should do before attending class. I will assume in many of my lectures that you have read this material. I separate the lecture topics into 4 relevant areas:

- General Software Engineering Knowledge and Exams (white)
- Software Design and Design Patterns (gray)
- Testing (light yellow)
- Emerging topics in Software Engineering (light blue)

Date	Lecture Topic	Reading
08/27 08/29	Class Introduction, expectations. Introduction to Software Engineering. Processes, Lifecycles.	Review your Object Oriented Concepts
09/03 09/05	UML, The Unified Modeling Language, Class diagrams.	An Introduction to UML Ch 1, 4, 5 UML 2.0
09/10 09/12	Introduction to Design Patterns (Strategy and Adapter patterns) UML Sequence diagrams	Ch 1, 7 Design Patterns (Strategy Pattern, Adapter Patterns) Ch 7 UML 2.0 HW1 due Sept. 12 during class

09/17 09/19	Observer Pattern UML Use case diagrams	Ch 2 Design Patterns (Observer Pattern) HW2 due XXX during class
09/24 09/26	Introduction to Agile techniques DevOps and SecDevOps	Handout
10/01 10/02	Virtualization, containers, the net, the cloud	Handout HW3 due XXX during class
10/08 10/10	Singleton and State Patterns	Ch 5 & 10 Design Patterns (Singleton and State Patterns)
10/15 10/17	Review and Mid Term	
10/22 10/24	Introduction to Testing	White & Black box testing, Mutation testing, Theory of testing.
10/29 10/31	Testing Criteria Introduction	The YoYo problem Amman and Offutt's Testing Criteria
11/05 11/07	Infrastructure Security (cryptography, key exchange, authentication, ssh, intrusion detection)	Handout Tool: http://designpatternfinder.codeplex.com/ HW4 due XXX during class
11/12 11/14	Design Concepts and Architectural Design	Component, Compositional Diagrams. Additional slides on component design (UML 2.0 Ch. 11,12)
11/19 11/21	Introduction to Microservices	Handout HW5 due XXX during class
11/26 11/28 Thanks Giving	Deployment pipeline	Handout Read about the Factory pattern on your own -- Ch 4 Design Patterns
12/03 12/05	Decorator and Iterator Patterns Exam Review	HW6 due XXX during class Ch 3 & 9 Design Patterns. Please evaluate this class https://www.cs.montana.edu/survey/index.php
12/09 – 12/13 Exams	Final Exam	TBD in Reid 103