

Meeting Times

- Tuesday, Thursday 12:15 - 13:30 in Reid 103.
- In person class.

Textbooks

- *Head First Design Patterns*, by Eric Freeman and Elisabeth 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 and only if you meet the 40 point minimum in your exams.
- Homework and exams will be graded on a straight scale. No curve.
- If you go 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 any material that is not yours must be cited.

Instructor and TA Information

- Faqeer ur Rehman. Office hours are on each Tuesday 10:00-12:00 p.m. at Student Success Center or by appointment.
- Email: matifkhattak@gmail.com, and faqeer.rehman@student.montana.edu .
- **TA:** Peng Zou, Email: peng.zou@student.montana.edu, Office hours: Each Thursday 3:10-5:10 p.m. at Student Success Center.

Policy

- You are responsible for any announcements made in class. These will be made either during lectures, or posted in D2L.
- **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.**
- Academic DISHONESTY results in an **automatic F** for you and the person you copy from. This should never be a problem, however, ****every year****, 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, ask me!
- You are expected to attend class in-person. 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.
- The class will be held in-person, so you must attend. They are not recorded.

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, Cloud, Security, 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
- Software Design and Design Patterns (gray)
- Testing (light yellow)
- Emerging topics in Software Engineering (light blue)

Date	Lecture Topic	Reading
<u>Week1 & Week2</u> 08/26 08/31 09/02	Class Introduction, expectations. Design and development of software, Object oriented concepts and principles.	Review your Object-Oriented Concepts HW1: due 09/07

	Introduction to Software Engineering. Processes, Lifecycles.	
<u>Week3</u> 09/07 09/09	UML, The Unified Modeling Language, Use case Diagrams, Activity diagrams, Class diagrams.	An Introduction to UML Ch 1, 2, 4, 5 UML 2.0 HW2: due 09/21
<u>Week4</u> 09/14 09/16	Introduction to Agile techniques, XP programming, SCRUM	Ch 1, 2 Design Patterns (Observer Pattern) Ch 7 UML 2.0
<u>Week5</u> 09/21 09/23	UML Review, UML Sequence Diagrams, Introduction to Design Patterns, Observer, Strategy	Ch 7 Design Patterns (Strategy) Ch 2 UML 2.0 HW3: 10/07
<u>Week6</u> 09/28 09/30	Different software Architectures, DevOps and SecDevOps	
<u>Week7</u> 10/05 10/07	Adapter Pattern, Virtualization, Containers Singleton and State Patterns	Ch 5 & 10 Design Patterns (Singleton and State Patterns) Ch 7 Design Patterns (Adapter Patterns)
<u>Week8</u> 10/12 10/14	Mid-Term (12 October) Cloud Computing, System optimization and scalability i.e., Horizontal and Vertical Scaling, DB sharding, Messaging queue etc.	
<u>Week9</u> 10/19 10/21	Mid-Exam Review Infrastructure Security (cryptography, key exchange, authentication, ssh, intrusion detection),	
<u>Week10</u> 10/26 10/28	Introduction to Testing, Different S/W testing techniques, test case generation techniques	White & Black box testing, Mutation testing, Theory of testing. HW#4 due : 04 Nov
<u>Week11</u> 11/02 11/04	Testing Strategies, Oracle problem, Metamorphic testing, Metamorphic relations, Differential Testing	The YoYo problem HW#6 due : 02 Dec
<u>Week12</u> 11/09 11/11	Compositional Diagrams Nov, 11 (Veteran's Day: No classes)	Component diagrams (UML 2.0 Ch 11)
<u>Week13</u> 11/16 11/18	Components Diagrams, Architectural Design, Organizing your Model: Packages Modeling Your Deployed System: Deployment Diagrams.	Component diagrams (UML 2.0 Ch 12, 13, 15)
<u>Week14</u> 11/23 11/25		No Class

<u>Week15</u> 11/30 12/02	Decorator pattern, Iterator pattern Formal Methods	Read about the Factory pattern on your own -- Ch 4 Design Patterns
<u>Week16</u> 12/07 12/09	Technical Debt / Review Lecture Final Exam (09 Dec 2021)	
<u>Week17</u> 12/14 12/16	Exam Review	Please evaluate this class https://montana.campuslabs.com/courseeval/