



Syllabus

Syllabus

COS 420: Object Oriented Design and Programming

Professor: Bruce MacLeod

www.cs.usm.maine.edu/~macleod

Room 222 Science Bldg.

email : macleod@usm.maine.edu

Tel: 780-4285

COS 420 Tutor:

Nickolas Littlefield

email: nickolas.littlefield@maine.edu

Course Description:

This course will focus on the construction of object oriented software. Students will learn conceptual models for organizing objects and object hierarchies, an object oriented design notation and the application of design patterns. These capabilities will be used to solve relatively complex problems as part of a software development teams.

Course Outcomes:

Successful completion of this course will provide students with the ability to :

- analyze a problem, and identify and define the computing requirements appropriate to its solution,
- to function effectively on teams to accomplish a common goal,
- to use current techniques, skills, and tools necessary for computing practice,
- to apply design and software development principles in the construction of software systems of varying complexity, and
- to understand different software development methodologies including Agile Development and the Rational Unified Process

Required Text:

- Applying UML and Patterns, Craig Larman, 3rd edition 2005, 0-13-148906-2

Prerequisite:

- Grade of C or better in either COS 350 or COS 360 or any COS 400 level course, or permission of instructor

Assignments:

There will be six assignments. All assignments are to be written in the Java programming language. There will be no provision for late assignments. Completion of the first assignment on time is necessary requirement for continuing in the course.

Grading:

- 20 % Exam
- 60 % Assignments (6)
- 20 % Class/Lab Participation (labs, homework, quizzes & participation in discussion)

Missed Lectures:

If you miss a lecture, you must complete the breakout room activity and the quiz (if there is one for that day) before the beginning of class. The breakout room activity(s) will be found in the lecture notes and the results from the completion of that activity should be posted to Brightspace in the Missed Lecture Assessment in Assignments . Late submissions will not be counted.

Links to other course materials.

Reading, assignment, and exam schedule

January

		1/18	Introduction, Software Development Methodologies, UML, Design Patterns Software for the course : <ul style="list-style-type: none">• Eclipse IDE for Java EE Devs.• Install Git• Sign up for a github account if you do not have one• Optional : Install UMLet in Eclipse (Help, Eclipse Marketplace, search UML)
1/23	Overview : Object Oriented Analysis and Design, Software Project Lifecycle AUP : Chapters 1-3 Lab : Github & Git HelloWorld tutorials	1/25	Project Inception & Use Cases AUP : Chapters 4-6
1/30	Other Requirements (AUP : Chapter 7) Agile Development Methodology Case Study if time permits Lab : Git Tutorial continued		

February

		2/1	Swing Tutorial (Get at least 5 examples running: submit brightspace, in class discuss) Swing Architecture
2/6	Assignment 1 Due, (ref : JavaDoc in Eclipse, JavaDoc Guidelines) Description of the Problem, Composition of Teams Overview of the Iterations, Teams Peer Evaluation Form Leader Peer Evaluation Form	2/8	Elaboration Phase : Domain models, AUP : Chapters 8&9

	Kick Off meeting for Teams Lab : Git Issues, Milestones, & Workflows		
2/13	Elaboration Phase : System Sequence Diagrams, Local Architecture AUP : Chapters 10-13	2/15	Object Design AUP : Chapters 14-16
2/20	No class : President's day	2/22	Iteration #1 Due, Project leader does a 10-15 min Presentations Composition of New Teams
2/27	Design Principles and Patterns AUP : Chapter 17		

March

		3/1	Object Design Examples AUP : Chapter 18
3/6	Read and investigate links in Classic Design Patterns Homework : Choose three patterns, one from each of the three categories (creational, structural, behavioral). In your own words, briefly describe the pattern you chose, describe another example (other than the one given by the site) and include a UML diagram of your example. Submit to Blackboard	3/8	Iteration #2 Due, Presentations Composition of Teams
3/13	Spring Break	3/17	Spring Break
3/20	Coding and Software Tests: Best practices AUP : Chapters 19-21 JUnit Testing Tutorial Homework : Provide a summary of Building the Perfect team Include your observations/thoughts on the article. Submit to Blackboard	3/22	GOF Patterns : AUP : Chapter 26
3/27	Java and Postgres Postgres Tutorial (first three sections) Quick JDBC Tutorial, Optional : More Detailed JDBC Tutorial JDBC Example Repository Link	3/29	Iteration #3 Due, Presentations Composition of Teams

April

4/3	Watch : Introduction to GCP for Students Watch : Getting Started with App Engine Sign up for Google Cloud Account (check email). Homework : Do GCP Postgres Quickstart Tutorial Submit output from exercise to Brightspace Optional, but quite useful : install pgadmin 4 Optional, but sometimes useful : install Postgres on local machine Reference : GCP Postgres Documentation Google Cloud Platform youTubes :	4/5	Servlet Tutorial Link to Repository with Sample Code (note : UN/PW in pom.xml file must be provided)
-----	---	-----	--

4/10	<p>JSP and Web Architecture</p> <p>JSP Youtube Video</p> <p>JSP Tutorial</p> <p>Homework :</p> <p>Do first four sections in GCP Java Web Tutorial</p> <p>Submit output to Brightspace</p> <p>Optional, but quite useful :</p> <p>Alternative JSP tutorial using examples</p> <p>install Tomcat on your local machine</p>	4/14	<p>Iteration #4 Due, Presentations</p> <p>Composition of Teams</p>
4/17	GCP, Postgres, Servlets, and JSP tutorial	4/19	Exam : Sample Questions
4/24	Project Meeting time	4/26	Project Meeting Time

May

		<p>Scheduled</p> <p>Final</p> <p>exam</p>	<p>Group Presentations. Attendance Mandatory</p>
--	--	---	--