PURDUE UNIVERSITY

CS 307: Software Engineering I

Fall 2026

Class:
MWF 12:30pm – 1:20pm, BRNG 1278

Course Web Page:
https://mandalore.cs.purdue.edu/~cs307/

Course Newsgroup:
https://edstem.org/us/join/3fTfbs

Instructor:
Dr. Jeff Turkstra, jeff@cs.purdue.edu, DSAI 1139E, 49-63088.

Office Hours:
TBD

Project Coordinators:
This course has one project coordinator. The names and email addresses for them are given below. All TA office hours are by appointment.

Mikyla Farnell Head TA	mfarnel@purdue.edu
Rohit Banerjee	baner121@purdue.edu
Ryan Broderick Jo	rjo@purdue.edu
Devansh Khandelwal	khanded@purdue.edu
Rachit Kumar	kumar800@purdue.edu
Christopher Kiyoshi Rodriguez	rodr1197@purdue.edu
Sayan Tapadar	stapadar@purdue.edu

Text:
Optional
Any up-to-date Software Engineering book (borrowed from a friend or from a library or in electronic format) will be fine. Make sure the book covers the Agile software development process – specifically, scrum.
Essentials of Software Engineering; Tsui, Karam, and Bernal; Jones & Bartlett, 2022
ISBN-13: 978-1284228991

Beginning Software Engineering; Rod Stephens; John Wiley & Sons, 2015
ISBN-13: 978-1118969144

Prerequisites:
Data Structures and Algorithms – CS 25100
Foundations of Computer Science – CS 18200
Programming in C – CS 24000
Problem Solving and Object-Oriented Programming – CS 18000
Programming proficiency is *absolutely* required

Course Outcomes:
A student who successfully fulfills the course requirements will have the ability to:
1. architect and design quality software
2. contrast and critique software lifecycle models
3. create software using current software engineering tools, methods, and concepts
4. work together on a team in a productive and professional manner
5. inspect and test software and documentation

Class Attendance:
You are expected to attend all classes. Attendance will be recorded for randomly selected class sessions. If you choose to attend class, please arrive in the classroom on time. You are expected to be quiet in class. If you must miss a class, you are responsible for procuring any material, information, handouts, announcements, etc., that you missed.

Preparation for Lectures:
You should try to read over the relevant pages in the course text before arrival. Additionally, you are expected to check your email and the course website regularly.

Here is the *tentative* lecture schedule:

Wk	Lec	Subject
1	1	Introduction, Software, and Software Engineering
	2	Software and Software Engineering
	3	Ethics
2	4	Ethics Continued, Software Quality
	5	Software Quality Wrap-up, Software Life Cycles
	6	Software Life Cycles cont., Scrum
3	X	Labor Day (No Lecture)
	7	Domain and Requirements Analysis
	8	Reusability, Client-Server Architecture
4	9	UML
	10	Design Patterns, Modeling Interactions and Behavior
	11	Software Design and Architecture

5	12	Architectural Patterns
	13	Revision Control and git
	14	git Wrap-up
6	15	Cohesion and Coupling, Users and Usability
	16	Inspection and Testing
	17	Software Testing
7	18	Software Testing cont.
	19	Project Management
	20	Risk Management
8	X	October Break (No Lecture)
	21	Peopleware
	X	No Lecture
9	22	Peopleware II
	23	Exam Review
10	X	**Final Exam**

Homework:
Homework assignments are assigned as needed to support class material and are due on the date given in the assignment. These are written assignments that are submitted electronically.

Quizzes:
There will be a number of quizzes during and after lecture. After lecture quizzes will be due no later than 24 hours after the end of lecture. Quizzes administered during lecture will be due in lecture. A score of zero will be given in case of no submission. The lowest grade will be dropped at the end of the semester. The quiz software used in this course requires a device capable of directly connecting to the PAL wireless network as well as providing accurate location information. Some devices have additional privacy features like location fuzzing, VPN-like proxying, etc that must be disabled. It is the responsibility of each student to properly configure their device.

Project:
The goal of the project component of this course is to establish a team of four to six people and engage in an intense software engineering effort following the scrum methodology. A project charter followed by a product backlog document will be generated. Subsequently, there will be three "sprints," each preceded by a planning document and proceeded by a retrospective. A review will take place during each sprint where a project coordinator will assess the efforts and accomplishments of the team. There will be a final project presentation at the conclusion of the semester. Documents submitted beyond established deadlines will incur a 10% penalty per day.

You may collaborate and work closely *only* with your teammates. The standard academic honesty policies apply to any inter-team communication and sharing.

Examination:
The exam will be closed book and closed notes. You must solve the exam problems yourself, without any help (knowing or unknowing) from any other student. You must not seek any knowledge in advance of the test questions (beyond that given in class) and must report any advance knowledge of the test questions by any student that you are aware of. You must not allow any other student access to your solutions during the exam.

Regrades:
Problems regarding grading of assignments and the exam must be resolved within **one week** after the graded work has been returned to you. It is your responsibility to pick up the graded work on time. Grades will not be modified after the one week period.

Late Submissions:
Homework assignments and project documents that are submitted after the specified deadline will incur a 10% reduction in score per day late.

Make-up Examination Policy:
Make-up exams will be given only in the **most extreme** circumstances and require certification for such circumstances. Eg, a medical doctor's statement certifying that the student is **unable** to attend the scheduled exam. Any travel (including interview trips), load from work or from other classes, failed alarm clocks, or simply not being able to make it to the exam will **not** be grounds for a make-up. If you have any recurring medical problems that may unexpectedly prevent you from making it to class or exams, please obtain a doctor's statement certifying your circumstance.

Documentation for Exceptional Events:
Any request for consideration due to an exceptional event requires an associated confirmation from the Office of the Dean of Students. E.g., in the case of a medical absence, a MEAP (Medical Excused Absence Policy) notification would be required. This is **in addition** to the requirements stated above.

Algorithmic Assistance:
Use of generative text tools in this course is permitted for the project portion *only*. Any time software is used to programmatically generate part of the project, it must be documented along with the inclusion of all prompts and model responses (i.e., a transcript of the interaction). Failure to provide documentation and attestation to the origins of third-party software in your project is a violation of the academic integrity policy below.

You are also encouraged to contemplate, prior to usage, any ethical and moral implications surrounding the use of models trained on material produced by individuals without compensation or consent.

Generative text tools *cannot* be used for homework assignments, quizzes, exams, etc.

Academic Integrity:
As a student at Purdue you are subject to the *Purdue University Student Code of Conduct*, which enjoins you to respect the highest standards of honesty and integrity. All work that you submit in this course must be your own; unauthorized group efforts are considered academic dishonesty. See the online brochure *Academic Integrity: A Guide for Students* for definitions and sanctions. Academic dishonesty is a serious offense which may result in suspension or expulsion from the University. In addition to any other action taken, such as suspension or expulsion, a **grade of F** will normally be recorded on the transcripts of students found responsible for acts of academic dishonesty. Students are encouraged to report academic dishonesty to the instructor directly, or to the Office of the Dean of Students.

You may discuss assignments in a general way with other students, but you may not consult anyone else's work. Among other ways to get an F, you are guilty of academic dishonesty if:

- You examine another student's solution to an assignment
- You allow another student to examine your solution to an assignment
- You fail to take reasonable care to prevent another student from examining your solution to an assignment and that student does examine your solution. For example, if you allow another student to check his/her email from your terminal while you step out of the room, you have failed to take reasonable care to prevent him/her from accessing your files.
- You submit an assignment that is not completely your own work
- You share results or notes during quizzes or exams

All work is subject to computer-based comparison and analysis. Do not con yourself into thinking that you can hide any collaboration. The risk of getting caught is too high, and the standard penalty is way too high.

If we find reason to believe that a student or team has cheated on any assignment, we may inform the student or team promptly, or we may decide to silently accumulate evidence against the student or team on later assignments.

Grading:
Final grades will be assigned according to the following *approximate* weighting:
Final Project Score – 65%
Exam – 25%
Homework and Quizzes – 10%

The grading breakdown for the project is as follows:

Project Charter – 5%
Backlog Document – 14%
Design Document – 14%
Sprint 1 Planning Document – 5%
Sprint 1 Review – 13%
Sprint 1 Retrospective – 2%
Sprint 2 Planning Document – 5%
Sprint 2 Review – 13%
Sprint 2 Retrospective – 2%
Sprint 3 Planning Document – 5%
Sprint 3 Review – 13%
Sprint 3 Retrospective – 2%
Final Project Presentation and Demo – 7%

These values summed represent the project score.

Peer Evaluations and Final Project Score:
The project score will be subject to a multiplier based on peer evaluation by your team members. Each person will evaluate his or her team members by distributing 10 * (n – 1) points, where n is the number of team members, among them. For a given member, a multiplier will be determined by summing the points given to them and dividing by 10 * (n – 1). This procedure will be done three times during the semester. The average of the resulting three multipliers, capped at 1.05, will then multiply the project score to generate the final project score. A deadline will be given for each aforementioned evaluation. Failure to meet the deadline will result in a decrease in your final multiplier by 0.05 each day past the deadline.

Should you choose to assign someone less than 8 points, there must have been explicit approval and an ongoing conversation with your project coordinator regarding that individual's performance issues. If this conversation is lacking, the number of points assigned to that individual will be raised to 8, deducted proportionately from all other team members' points, and a mandatory meeting between you and the instructor will be scheduled.

Finally, failure to provide a reasonable rationale for your point assignments will result in a reduction of your multiplier by 0.05.

Questions and Answers:
Questions of general interest should be posted on the course newsgroup. Answers will be posted as soon as possible. Project questions should be directed to the appropriate project coordinator via email. Answers will be sent to you directly. If you need to contact a specific TA or instructor, send email to that individual or go see him/her during office hours.

Modifications:
This syllabus may be modified at any time with notification.

**** As an interesting side note, a significant portion of this syllabus is copied from Dr. Dunsmore, Dr. Hosking's, Dr. Brylow's, and Dr. Hu's policy pages from previous semesters. One of the major differences between plagiarism and proper reuse is giving credit where credit is due. ****