Course Title: CSI 3370 -11746 - Software Engineering (4 credits)

Required Text Book:
Software Engineering: Modern Approach, Second Edition. By Eric J. Braude and Micheal E. Bernstein
Available for online purchase through the

Term: Spring 2025 Semester (01/06/2025– 04/26/2025)
Lecture Days, Time (Location): MW, 3:30 – 5:17; MCS 102

Instructor: Destiny Anyaiwe, PhD **Email:** oanyaiwe@oakland.edu

Course Policy on Using AI Platforms:
In this course, students are encouraged to leverage AI platforms and tools (e.g., ChatGPT, WolframAlpha, NetworkX, etc.) as supplementary resources to enhance their understanding of topic/subject concepts and their applications. These tools can provide explanations, examples, and insights that support your learning journey.

However, **it is crucial to maintain academic integrity** and use AI responsibly. The following guidelines (not exhaustive) clarify acceptable and unacceptable uses of AI tools:

Acceptable Use of AI

- Using AI to clarify definitions, concepts, algorithms, and problem-solving techniques.
- Asking AI for alternative explanations or examples related to course material.
- Using AI to generate ideas for your assignments, final project, or real-world applications of Software Engineering
- Consulting AI for syntax tips or debugging suggestions when writing codes, and algorithms.
- Asking AI to simplify or expand upon challenging topics discussed in class.

Unacceptable Use of AI

- All submitted assignments, projects, and exams must be your original work not AI-generated.
- Using AI to directly solve homework problems, exams, or project tasks without attempting to solve them yourself, thereby bypassing learning.
- Excessively using AI in place of your critical thinking and problem-solving efforts.

Consequences of AI Misuse

The consequences of cheating or plagiarism, including the misuse of AI are a zero score and penalties outlined in the university's academic integrity policy.

Catalog Course Description:

Software engineering is the branch of computer science that creates practical, cost-effective solutions to computing and information processing problems, preferentially by applying scientific knowledge and developing software systems in the service of mankind. This course covers the fundamentals of software engineering, including understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development, and the application of engineering tools.

Course Outcomes:

The course learning outcomes are aligned with the weekly course teaching and learning goals and assignments. The outcomes will be evaluated through assignments, assessments, and other methods throughout the course. On completion of this course, you will be able to:

1. Describe software processes and how to differentiate between them
2. Determine how to elicit requirements from a client and specify them
3. Design in the large, including principled choice of software architecture, the use of modules and interfaces to enable separate development and design patterns.
4. Practice good coding practices, including documentation, contracts, regression tests, and daily builds.
5. Effectively implement version control, configuration management, unit/regression testing, issue tracking, and debugging tools
6. Create a project plan
7. Create and analyze design models
8. Apply engineering tradeoffs
9. Work effectively in teams
10. Put effective software processes into practice
11. Handle large legacy systems and frameworks such as JIRA, GoogleDocs, Github, GUIs, EJBs, and Eclipse
12. Communicate with clients to elicit requirements and manage expectations

Course Schedule:

Dates	Modules	Outcomes	Topics / Readings	Assignments Due
Week **One**	Module 0 Course Intro		Course Orientation Post Course Syllabus Student Course Testimonial Video and software failure videos.	Student Introduction & Discussion. Discussion guideline: Students pin down their concerns and interests using text, videos, paper referencing, online news, and interests going forward.
Week **Two – Three**	Module 1 Introduction to software engineering	1; 3; 6; 7; 8	Chapters 1 – 3	Discussion One Assigned: Initial Post-Due Quiz #1 Assignment #1
Week **Four -Five**	Module 2 **A)** Agile Software Processes **B)** Quality in Software Process	1 – 12	Topics/Chapter 4 – 5	Discussion One Responses Quiz #2 Assignment #2
Week **Six**	Module 3 Configuration Management	3 – 5 & 9	Topics/Chapter 6 Git+Github intro	Discussion Two Assigned: Initial Post-Due Quiz #3
	Git+Github intro			Milestone 0: Group assignment & topic selection
Week **Six – Seven**	Module 4 Software Requirements	1;2;3;4;8; 10; 11;12	Topics/Chapter 12 -13	Discussion Two Responses Quiz #4 Assignment #3
Week **Eight – Nine**	Module 5 Software Requirements continues	1;2;3;4;8; 10; 11;12	Topics/Chapter 14	Discussion Three Assigned: Initial Post-Due Quiz #5 Milestone 1: Project Progress update
Week **Ten**	Module 6 **OverFlows**	Re-enforcing of 1 – 12	Previous Topics Review: Discussion Open for general questions and comments	Discussion Three Responses Mid-Term Exam
Week **Eleven Twelve**	Module 7 Software Design	1;3;5;7;8; 10;11;12	Topics/Chapter 15-16	Discussion Four Posted: Initial Post Due Quiz #6 Assignment #4
Week **Thirteen – Fourteen**	Module 9 Implementation & Testing	4 – 8 & 10	Topics/Chapters 22-26	Discussion Five Assigned: Initial post Quiz # 8 Milestone 3 Assignment #5

Week **Fifteen**	Module 10 Finals Week	1 – 12	Topics/Chapter Course Revision Project Presentation & submissions Final Exam	Discussion Five: Response Post Open Discussion Forum Milestone 4: Project Due Zoom Presentation

Student Assessment:

The course has 8 quizzes (worth 400 points), 5 homework assignments (worth 100 points), and 10 graded discussion topics totaling 100 points. A capstone project (250 points), mid-term (50 points), and final exam that is worth 100 points. Letter grades are awarded based on the total number of points achieved. The quizzes will be available and taken on Canvas.

Assessments:

Mid-term Exam: 20%, Final Exam 2: 25%, Assignments: 20%, Project: 30%, and Class Participation: 5%.

Grade Scale:

A+ >= 95, A >= 92, A- >= 90, B+ >= 85, B >= 82, B- >= 80, C+ >= 75, C >= 72, C- >= 70, D+ >= 65, D > 62, D>= 60, E<=59.

Projects

- There will be a capstone project. The project rubric will be made available in the project description and it will also be used to evaluate the projects. Milestones will be used to assess project progress.

Quizzes

- There will be 8 (40 – 50 minutes long) quizzes. Each Quiz will cover materials for the previous week/module

Late Submissions

- Late submissions are not allowed