

# Syllabus Fall 2022

## CSCE 247: Software Engineering

### Course Details

Course: CSCE 247  
Credits: 3  
Times:  
Section 1: 8:30am – 9:45am TR (Innovation Center 1400)  
Section 2: 1:15pm – 2:30pm TR (300 Main st. B110)

### Contact Information

Instructor: Portia Plante  
Email: [pplante@cse.sc.edu](mailto:pplante@cse.sc.edu)  
Office: 2275 Storey Innovation Center, & Online  
Office MW: 10:00am – 11:30am  
Hours: TR: 10:15am – 11:15am  
On demand through zoom or MS Teams ([pplante@cse.sc.edu](mailto:pplante@cse.sc.edu))  
Email to schedule a meeting

### Contact Information - TAs

TA Name: Tyler Barrett  
Email: [tylertb@email.sc.edu](mailto:tylertb@email.sc.edu)  
TA Name: TBD  
Email: [TBD](#)

### Academic Bulletin Description

Fundamentals of software design and development; software implementation strategies; object-oriented design techniques; functional design techniques; design patterns; design process; source control; testing.

### Course Description

Software engineering is concerned with the development and evolution of high-quality software systems in a systematic, controlled, and efficient manner. Software engineers are concerned with safety and reliability of the product as well as the cost and schedule of the development process. The lectures and the group projects will cover all aspects of the software life cycle, from development team management, problem specification and analysis, system design techniques, implementation and documentation practices, testing, to maintenance and evaluation of the final product.

### Prerequisites

C or better in CSCE 146

## Learning Outcomes

All learning outcomes in this Distributed Learning course are equivalent to face-to-face (F2F) version of this course.

1. Distinguish between software development processes and choose an appropriate process for a particular project, including the selection of appropriate source control and project management tools.
2. Elicit requirements and create a requirements specification document.
3. Develop software architectural models and analyze how control and data flow through a system.
4. Apply the principles of object-oriented software design, including how to describe and model the structure of a system.
5. Apply software design patterns.
6. Apply the fundamentals of requirements-based and structure-based software testing and the accompanying test selection methods.
7. Apply human computer interaction theory and design principles.

## Recommended Textbooks

Head First Design Patterns. (1<sup>st</sup> Edition), Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson., O'Reilly Media

All readings/materials comply with copyright/fair use policies.

## Technology Requirements

- A reliable internet connection
- A speaker, microphone and webcam for effective online meetings
- A Java IDE of your choice (Eclipse, IntelliJ, ...)
- Git bash (Instructions for downloading and using will be provided)
- LucidCart or a similar online UML editing tool
- Proto.io or a similar prototyping tool

## Course Delivery

This course will be delivered in-person.

**Course Materials:** All course materials can be found on [dropbox.cse.sc.edu](https://dropbox.cse.sc.edu). This includes, videos, assignments, and tests. This course will not be using Blackboard. You can sign into dropbox with your traditional uofsc account.

**Student-to-Instructor (S2I) Interaction:** Students will interact with the instructor during in-class discussions, and group meetings.

**Students-to-Student (S2S) Interaction:** Students will interact during in-class discussions, and in groups.

**Student-to-Content (S2C) Interaction:** Students will engage with course content by completing programming assignments, completing a semester long team project, and completing participation activities.

## Topical Outline

- Introduction to SE/Principles
- Requirements Specification
- Human Computer Interaction
- Project Management
- Software Architecture
- Design (OO)
- Implementation
- Testing
- Source Control

## Deliverables:

### **Design Pattern Assignments:**

Students will complete a small Java development project to implement the respective design pattern, to help better understand their value in context. A detailed description of each assignment is in dropbox.

- Strategy Design Pattern
- Observer Design Pattern
- Decorator Design Pattern
- Factory Design Pattern
- Iterator Design Pattern
- Singleton Design Pattern
- Command Design Pattern
- Adapter Design Pattern
- State Design Pattern

### **Git Exercises:**

Two short, hands-on exercises to familiarize students with the git dev environment before utilizing it in their team implementation. A detailed description of each assignment is in dropbox.

- Git Branching Exercise
- Team git exercise

## Semester Team Project:

During the semester long project, students will complete a medium-scale development project in teams of 3 to 4 students. They will begin by eliciting their projects requirements from their customer. They will then create a development design document to outline their development process. They will then implement and test their product, creating a Java based console program. Finally, they will make a prototype of a graphical version of this program.

- Requirements Document Part 1
- Requirements Document Part 2
- Development Design Document
- Implementation, SCUM board, and Video Presentation
- Testing
- Low Fidelity Prototype
- High Fidelity Prototype and Video Presentation

## Test Schedule

Tests will be available for a 24 hour period beginning at 12am – 11:59pm on the specified day. Tests will be available online through Dropbox. Once the test is begun students will have 2 hours to complete it. The tests are open book. Students are allowed to use all resources.

Midterm: October 4<sup>th</sup> online  
Final Exam: Exams will be held online for both sections.  
Section 1: Tuesday December 6<sup>th</sup>  
Section 2: Thursday December 8<sup>th</sup>

## Grading Policy

Design Patterns:	15%
Git Exercises:	2%
Requirements P1:	3%
Requirements P2:	10%
Development Design Document:	10%
Implementation, SCRUM board and	20%
Video Presentation:	
Testing:	10%
Low Fidelity Prototype:	2.5%
High Fidelity Prototype:	7.5%
Midterm (Test 1):	10%
Final (Test 2):	10%

Students must contribute fairly to their team project to receive a passing grade. Students will be surveyed regularly to ensure that all team members are contributing fairly to team projects.

The grade is calculated using the standard curve:

Final Grade Range	Reported Grade
90-100%	A
87 – 89.99%	B+
80 – 86.99%	B
77 – 79.99%	C+
70-76.99%	C
67 – 69.99%	D+
60-66.99%	D
<60%	F

## Grade Discussion

Questions about any grades in this class must be addressed within 1 week of work being returned.

## Late Work

Homework assignments are due at the time listed on Dropbox. A 25% deduction per day late will be applied to assignments. Submitting all assignments is a necessary condition for passing this class.

## Weekly Schedule

Week	Description	Due Dates
Day 1 <b>Aug 18<sup>th</sup></b>	Introduction to the course: - Syllabus - Introduction to Software Engineering - Inheritance and Interfaces - Setup Environment	
Week 2 <b>Aug 22<sup>nd</sup> - 28<sup>th</sup></b>	- Personas - Competitive Analysis - Strategy Design Pattern - JavaDoc - Intro to VS Code IDE - Introduction to version control via git	
	<b>Assignments:</b> - Strategy Design Pattern - Requirements Part 1	Aug 28 <sup>th</sup> 11:59pm

Week 3 <b>Aug 29<sup>th</sup> - Sept 4<sup>th</sup></b>	<ul style="list-style-type: none"> <li>- Software Requirements</li> <li>- More requirements</li> <li>- Requirements elicitation</li> <li>- Observer Design Pattern</li> </ul>	
	<b>Assignments:</b> <ul style="list-style-type: none"> <li>- Observer Design Pattern</li> <li>- Prepare for Elicitation</li> </ul>	Sept 4 <sup>th</sup> 11:59pm
Week 4 <b>Sept 5<sup>th</sup> - Sept 11<sup>th</sup></b>	<ul style="list-style-type: none"> <li>- Client Interaction Session to gather requirements</li> <li>- Business use cases</li> <li>- Decorator design pattern</li> </ul>	
	<b>Assignments:</b> <ul style="list-style-type: none"> <li>- Decorator Design Pattern</li> <li>- Requirements Part 2</li> </ul>	Sept 11 <sup>th</sup> 11:59pm
Week 5 <b>Sept 12<sup>th</sup> - Sept 18<sup>th</sup></b>	<ul style="list-style-type: none"> <li>- Software Design Fundamentals</li> <li>- Software Architecture</li> <li>- Git Branching</li> <li>- Factory Design Pattern</li> </ul>	
	<b>Assignments:</b> <ul style="list-style-type: none"> <li>- Factory Design Pattern</li> <li>- Git Branching</li> </ul>	Sept 18 <sup>th</sup> 11:59pm
Week 6 <b>Sept 19<sup>th</sup> - Sept 25<sup>th</sup></b>	<ul style="list-style-type: none"> <li>- UML Class Diagrams</li> <li>- UML Sequence Diagrams</li> <li>- Git – working with teams</li> <li>- Iterator Design Pattern</li> </ul>	
	<b>Assignments:</b> <ul style="list-style-type: none"> <li>- Iterator Design Pattern</li> <li>- Team Git Branching</li> <li>- Design Document Diagrams</li> </ul>	Sept 25 <sup>th</sup> 11:59pm
Week 7 <b>Sept 26<sup>th</sup> - Oct 2<sup>nd</sup></b>	<ul style="list-style-type: none"> <li>- UML Class Diagrams Extended</li> <li>- UML Class Diagrams accessing a data store</li> <li>- Singleton Design Pattern</li> <li>- OOP Design Patterns and principles</li> <li>- Midterm Prep</li> </ul>	
	<b>Assignments:</b> <ul style="list-style-type: none"> <li>- Singleton Design Pattern</li> <li>- Design Document Diagrams Extended</li> </ul>	Oct 2 <sup>nd</sup> 11:59pm
Week 8 Day1 <b>Oct 4<sup>th</sup></b>	<b>Assignments:</b> <ul style="list-style-type: none"> <li>- Midterm Exam</li> </ul>	Oct 4 <sup>th</sup>
Week 8 Day2 <b>Oct 6<sup>th</sup> - Oct 9<sup>th</sup></b>	<ul style="list-style-type: none"> <li>- Command Design Pattern</li> <li>- Code organization with a driver</li> <li>- Parsing JSON</li> <li>- Refactoring and code evolution</li> </ul>	
	<b>Assignments:</b> <ul style="list-style-type: none"> <li>- Command Design Pattern</li> </ul>	

	- Design Document Complete	Oct 9 <sup>th</sup> 11:59pm
Week 9 Day 1 <b>Oct 11<sup>th</sup></b>	- Refactoring Example - Introduction to SCRUM - Creating a SCRUM Board	
	<b>Assignments:</b> - SCRUM Board	Oct 12 <sup>th</sup> 11:59pm
<b>Oct 13<sup>th</sup> - Oct 14<sup>th</sup></b>	Fall Break	
Week 10 - 12.5 <b>Oct 17<sup>th</sup> - Nov 2<sup>nd</sup></b>	- Project Implementation - SCRUM Meetings (Each group will have two 10 minute synchronous meetings a week, which will be scheduled in advance)	
	<b>Assignments:</b> - Implementation and Presentation	Nov 2 <sup>nd</sup> 11:59pm
Week 12.5 - 14 <b>Nov 3<sup>rd</sup> - Nov 13<sup>th</sup></b>	- Testing Basics - Test Automation - Structural Testing - JUnit Testing - Template Design Pattern	
	<b>Assignments:</b> - Testing	Nov 13 <sup>th</sup> 11:59pm
Week 15 <b>Nov 14<sup>th</sup> - Nov 20<sup>th</sup></b>	- Sketching and Prototyping - Creating a low-fi Prototype - Usability Testing - State Design Pattern	
	<b>Assignments:</b> - State Design Pattern - Low Fidelity Prototype	Nov 20 <sup>th</sup> 11:59pm
Week 16 <b>Nov 21<sup>th</sup> - Nov 29<sup>th</sup></b>	- Creating a high fidelity prototype - Visual Design Principles - Adapter Design Pattern - Final Prototype and presentation	
	<b>Assignments:</b> - Adapter Design Pattern - High Fidelity Prototype - Prototype In class presentation	Nov 30 <sup>th</sup> 11:59pm Dec 1 <sup>st</sup> In Class
Final Exams	Section 1 Section 2	Dec 6 <sup>th</sup> Dec 8 <sup>th</sup>

## Attendance Policy

Attending in-person will not be formally checked. There will be mandatory meetings between group members and the instructor in-person (or online in the event of sickness). Attending these meetings will help ensure that you are actively participating in your team project which is a requirement of the class.

## Turnaround Time

Instructor will reply to all feedback in a reasonable amount of time, and the same expectations are made for students. Expectations are listed below.

**Communication:** Responses to email communication and questions will be given within 24 hours

**Assignment Grading:** Grades for assignments will be returned within 72 hours of due date.

**Test Grading:** Grades for tests will be returned within 72 hours of test completion time.

## Syllabus Change Policy

This syllabus is a guide and every attempt is made to provide an accurate overview of the course. However, circumstances and events may make it necessary for the instructor to modify the syllabus during the semester and may depend, in part, on the progress, needs, and experiences of the students. Changes to the syllabus will be made with advance notice.

## Policies and Procedures

This section contains some general rules that will be enforced during this course. Please review these guidelines carefully. The course is governed by the policies and procedures of the university (<http://www.sc.edu/policies/ppm/staf625.pdf>). Violations of this code can result in actions varying from a failing grade to expulsion from the university.

## Academic Integrity

University policies and procedures regarding academic integrity are defined in in policy STAF 6.25, Academic Responsibility - The Honor Code (see <http://www.sc.edu/policies/ppm/staf625.pdf>). Prohibited behaviors include plagiarism, cheating, falsification, and complicity. All potential Honor Code violations will be reported to the Office of Academic Integrity, which has the authority to implement non-academic penalties as described in STAF 6.25. Academic penalties for Honor Code violations in this course range from a zero on the assignment to failure of the course.



## CSCE 247 Academic Integrity

Examples of Academic Integrity violations in CSCE 247 include:

- Copying another students programs or files
- Copying a program or file from the internet

You may reference an existing program from the internet with proper citation, assuming you do not simply copy it.

You will be required to sign an honor code statement when submitting assignments and exams.

## Accommodating Disabilities

Reasonable accommodations are available for students with a documented disability. If you have a disability and may need accommodations to fully participate in this class, contact the Student Disability Resource Center: 777-6142, TDD 777-6744, email [sasds@mailbox.sc.edu](mailto:sasds@mailbox.sc.edu), or stop by Close-Hipp Suite 102. All accommodations must be approved through the Student Disability Resource Center.