

SENG 6230: Software Engineering Foundations

Fall 2017

Class Meeting	Tuesday and Thursday, 2:00pm – 3:15pm Brewster Building, Room B-203 (Global Classroom)
Instructor	Dr. Mark Hills
Office	Science & Technology Building, C-110
Office Hours	Tuesday 3:30pm – 5:00pm Wednesday 10:00am – 12:00pm Thursday 3:30pm – 5:00pm
Phone	252-328-9692
Email	hillsma@ecu.edu (response within 24 hours during the week, possibly longer on weekends)
Course web page	https://blackboard.ecu.edu
Slack	https://ecu-seng-6230.slack.com

Course Summary

The catalog description for this course is as follows:

Software project development using software engineering principles and current software development techniques.

This course provides the practical and theoretical knowledge of software engineering used as a foundation for both software engineering research and practice. Students will learn the processes, methodologies and tools used during the complete life cycle of professional software development, allowing them to begin using state-of-the art software development techniques that will aid in the success of their software development and research projects. Students will also begin to explore current software engineering research through relevant research papers and a course writing assignments. Students are required to complete both a team project and an individual writing project over the course of the semester.

Prerequisites

The only prerequisites for this course are enrollment in either the Computer Science or Software Engineering graduate programs and the consent of the instructor. You are expected to be familiar with Java at the level of a student that has successfully completed an introductory computer science sequence or SENG 5000. You may opt to use a different language on your project (if your project team opts to do so, this is a group decision), but this gives us a shared programming language for examples and a shared baseline to build from in terms of development knowledge and concepts (e.g., the distinction between a class and an object of that class). If you are familiar with a similar language, especially C#, adapting to Java should be fairly straight-forward. If you are not familiar with Java, please read, and go through the exercises in, a book like

Schildt's "Java: A Beginner's Guide". There are also a number of online resources you can look to, including videos on the Lynda.com tutorial service (available to all ECU students). If in doubt, ask me.

Learning Outcomes

After taking this course, you should be able to:

- Develop software systems using state-of-art development techniques and tools
- Evaluate and choose processes for the development of software systems
- Plan and manage software projects
- Analyze and design software systems using object-oriented methods
- Perform software testing
- Deliver and deploy software systems
- Evolve and maintain software systems

Tools and Applications

The following tools and applications may be used in this course:

- UML Modeling: NoMagic MagicDraw or Rational Software Architect
- Gantt Charts: Microsoft Project
- Development Environments: Microsoft Visual Studio (C# and .NET), Eclipse (Java and J2EE), Android Studio (Java/Android), XCode (Swift/iOS), PHPStorm (PHP), or other language-specific IDEs
- Version Control: Git and GitHub
- Continuous Integration: Travis-CI
- Project Task Management: Atlassian Jira
- Testing: JUnit (Java), NUnit (.Net), PHPUnit (PHP), or other language-specific unit testing frameworks
- Application Deployment: Google Cloud Services (if the project is a web-based system or uses some sort of web or database backend)

Some of these tools are open-source, while others are commercial tools that you can access without charge. Links to these tools will be provided on Blackboard as needed.

Textbooks

The required textbook for this course is *Software Engineering (10th Edition)*, by Ian Sommerville. This book is available through Amazon.com at <https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038> and should also be available through the ECU campus bookstore.

Other helpful material, including references to books, conference or journal articles, tutorials on the web, and videos will be posted as the course progresses.

Topics

Topics covered in this course include:

- The nature of software and software engineering practice
- Software process models
- Software requirements management
- Software requirements documentation
- Traditional software design concept
- Object-oriented software design concepts
- Modeling with Unified Modeling language (UML)
- Software testing strategies
- Software testing documentation
- Project management concepts

We will cover the following chapters in the textbook that are related to these topics:

- Introduction to Software Engineering (Chapter 1)
- Traditional and Agile Software Processes (Chapters 2 and 3)
- Project Management and Planning (Chapters 22 and 23)
- Requirements Engineering (Chapter 4 and 5)
- Software Design and Architecture (Chapters 6 and 7)
- Implementation and Configuration Management (Chapters 7 and 25)
- Software Testing and Quality Management (Chapters 8 and 24)
- Software Evolution (Chapter 9)

As time permits, we may also look at focus areas such as service-oriented software engineering (Chapter 18) and software reuse (Chapter 15).

Rough Deliverable Schedule

The course will roughly proceed as follows. These dates are subject to change, so please check Blackboard for current dates on these items.

Dates	Items of Note
August 24	Groups Assigned by Random Assignment
August 31	Group Deliverable: Project Abstract due Writing Assignment: Topic Selected
September 12	Group Deliverable: Project Planning, Risk Management
September 14	Writing Assignment: Annotated Bibliography Part 1 Due
September 26	Group Deliverable: Requirements Specification
September 28	Writing Assignment: Draft 1 Due
October 5	Midterm Exam
October 6	Feedback on Writing Assignment Draft 1 Due

October 10	NO CLASS (Fall Break)
October 12	Writing Assignment: Annotated Bibliography Part 2 Due
October 26	Writing Assignment: Draft 2 Due, with Review Responses
November 3	Feedback on Writing Assignment Draft 2 Due
November 19	Group Deliverable: Design Artifacts/Diagrams
November 16	Writing Assignment: Due, with review responses
November 21	Group Deliverable: Testing Approach
November 23	NO CLASS (Thanksgiving)
November 28	Group project presentations, day 1
November 30	Group project presentations, day 2
December 4	Group deliverable: final project report, all code
December 12	Final Exam

The group project deliverables before the final project report are worth credit, but are also an opportunity to provide feedback. Any issues pointed out with these deliverables should be fixed for the final project report.

Exams

The midterm exam for the course will be on **Thursday, October 5th**, from **2pm – 3:15pm** in our normal classroom. More details about the exam will be available closer to the exam date. The exam will be a closed book exam, but you will be allowed to bring one page (letter size, front and back) of hand-written notes..

The final exam for the course will be on **Tuesday, December 12th**, from **2pm – 4:30pm** in our normal classroom. More details about the exam will be available closer to the exam date. The exam will be a closed book exam, but you will be allowed to bring one page (letter size, front and back) of hand-written notes. We will review for the exam on Thursday, November 30th, which is the last day of class.

If you are taking the course online, you must have a proctor for the midterm and for the final exam. You must use the University of North Carolina Proctoring Network. More information can be found at: <http://online.northcarolina.edu/exams/overview.htm>

Grading

Students will be evaluated based on the combination of class activities. The final grade will be assessed with the following criteria:

Grading	
A	≥ 90
B	≥ 80
C	≥ 70
F	< 70

This grade is based on the following relative weights of the various activities:

Weighting	
Midterm Exam	20%
Final Exam	20%
Weekly Quizzes	10%
Writing Assignment	20%
Group Project	30%

The group project will be graded based on the following point distribution (viewing the 30% as 30 points):

1. Select and follow a software process model to develop the project. (3 points)
2. Develop a project schedule and project management plan, monitor project progress (including risk planning and management). (3 points)
3. Define system requirements including functional requirements and non-functional requirements. (5 points)
4. Define system design models using UML. (5 points)
5. Implement the system. (5 points)
6. Develop test suites for unit test and system test, and document test results. (5 points)
7. Document (final report) and present the project. (4 points)

The grade breakdown for the writing assignment can be found in the details on the assignment.

Starfish

This course uses the Starfish system to provide you with information on your performance within the course. For more information, please see <http://www.ecu.edu/cs-acad/advising/upload/Starfish-Student-Getting-Started.pdf>.

Student conduct

Smoking is not permitted in classrooms. Please turn off telephones while in class. Laptops and tablets can be used for taking notes, but should not be used for other work (or recreational browsing, playing games, etc).

Students are expected to abide by the university's Student Honor Code. The homework that you do is a critical part of your education. Each student is expected to do his or her own work, except where teamwork is explicitly allowed or required. That does not mean you are not allowed to discuss your ideas with other students. Working in groups can be beneficial, and I encourage you to talk through ideas with other students. But outright copying is plagiarism and is unacceptable. Students who copy other students' work, or who allow their work to be copied, or who copy their work from other sources, such as the internet, are violating the ECU academic integrity policy. Not only that, if you are copying your answers instead of doing the work yourself, you are essentially

missing the entire point of this course, which will come back to haunt you when you don't know this material at a future employer.

Other potential academic integrity violations are cheating, falsification, multiple submissions of the same work in different classes, and attempts at any of these violations. Please see http://www.ecu.edu/cs-studentlife/policyhub/academic_integrity.cfm for more details.

Academic integrity violations can result in a grade penalty up to and including an F for the course.

Other Policies

No incompletes will be issued in this course except for extraordinary circumstances, and even then only if you are nearly done already, and have done work of acceptable quality, so that you have a realistic chance to pass the course.

All group project deliverables, writing assignment deliverables, and quizzes are due by the posted due date and time. Late submissions will not generally be accepted. If for some reason you are not able to complete the assignment on time, you must contact me directly with an explanation and request an extension before the deadline. If something comes up and you are having trouble keeping up with the class, talk to me right away, ***don't wait until the end of the semester!***

Course participation is an important part of the course. If you do not participate you will make it harder to have the kinds of discussions we need to make the class interesting. Please read any assigned readings in a timely fashion, do the assignments promptly, and come to class prepared to talk.

Success in the class is directly correlated with class attendance, so I highly recommend that you attend and actively participate. If for some reason you cannot attend, please let me know – my expectation is that you will watch the lecture online and ask me questions about the material if you have any. For online students, I recommend that you watch the lecture the day it is given and send any questions before the next class session (so I can address them in class). Falling behind will make the course more difficult than it would otherwise be. I will be taking attendance at regular points in the class for my own records.

All code, test scripts, and other software artifacts for your group project must be stored in GitHub (**this is not optional**). I will not accept code submitted through Blackboard or emailed to me. If you have questions about your code, check it in to the related GitHub repository, that way I can easily look at it.

Weather emergencies

In the event of a weather emergency, information about ECU can be obtained through the following sources:

ECU emergency notices <http://www.ecu.edu/alert>
ECU emergency information hotline 252-328-0062

Students with disabilities

East Carolina University seeks to comply fully with the Americans with Disabilities Act (ADA). Students requesting accommodations based on a disability must be registered with the Department for Disability Support Services located in Slay 138 ((252) 737-1016 (Voice/TTY)).

For more information, please see <http://www.ecu.edu/cs-studentlife/dss/>.

Retention Requirements

Academic requirements for retention have changed. Please be aware of the following new GPA requirements. Please discuss the retention requirements, entrance to major requirements, and your goals with your academic advisor.

GPA Hours at ECU (identified in Transcript in Banner Self Service) plus transferred credit hours	"Old" Retention Requirement All courses taken at ECU	New Retention Requirements Effective with Fall 2011 grades All courses taken at ECU
1-29 semester hours	1.6 GPA	1.8
30-59 semester hours	1.8 GPA	1.9
60-74 semester hours	1.9 GPA	2.0
75 or more semester hours	2.0 GPA	2.0

Caveats

Occasionally, it may be necessary to revise this syllabus due to extenuating circumstances. I reserve the right to revise this syllabus if the need arises. If I do so, I will provide you with advance notice.