COMP 4081: Software Engineering
Fall 2019

Monday, Wednesday 12:40–2:05 p.m.
Comm Fine Arts Bldg 105

https://memphis-cs.github.io/comp-4081/

Please send all emails to this address:
comp4081-instructors@googlegroups.com

Instructor:	Dr. Scott Fleming <Scott.Fleming@memphis.edu>
Office Hours:	Mon, Wed 2:15–3:00 p.m. or by appointment
Office:	Dunn Hall 303
Teaching Assistant:	TBA <tba@memphis.edu>
Consulting Hours:	By appointment

1 Catalog Description

COMP 4081 - Software Engineering (3)
Scope of software engineering; software life cycle models; software process; team organization; requirements analysis and design methodologies; metrics, inspections, testing strategies and maintenance; software risks; professional and ethical responsibilities. Computer Science majors should plan to take COMP 4882 during the following spring semester. It is recommended that students take COMP 3115 before taking this course. PREREQUISITE: COMP 2150, permission of instructor.

2 Topics

The course will emphasize the following topics (a subset of the knowledge areas in the *Guide to the Software Engineering Body of Knowledge*):

- **Software Requirements**, including elicitation, specification, and validation
- **Software Design**, including principles, methods, patterns, and notations
- **Software Construction**, including tools, platforms, and APIs
- **Software Testing**, including levels, techniques, and measures
- **Software Maintenance**, including types and cost
- **Software Configuration Management**, including version control
- **Software Engineering Management**, including planning and estimation
- **Software Engineering Process**, including life-cycle models and assessment

3 Course Outcomes

The following are some key learning goals that students in the course are expected to achieve:

1. Use modern software development tools (e.g., Git) to perform effective configuration management.
2. Design and model software using modern modeling languages and notations (e.g., UML).

3. Convey ideas orally through presentations to peers, clients, and faculty.
4. Gather and specify software requirements (e.g., as user stories).
5. Apply a modern software engineering process (e.g., Scrum) to the development of a software project.
6. Produce high-quality software-development artifacts for each phase of the development cycle.
7. Apply modern software testing tools and techniques (e.g., black- and white-box testing).
8. Develop software in collaborative teams.
9. Plan work and to distribute tasks among team members to maximize team productivity.

4   Required Equipment

Students will be required to bring a laptop to all lectures.

System Requirements: Capable of running Linux virtual machine (VirtualBox). (Hardware must support virtualization.)

5   Required Textbooks

No textbooks are required for this course.

6   Evaluation

Grading weights are as follows:

- 35% Skills Tests
- 25% Knowledge Tests
- 40% Team Project
  - 20% Team Achievement
    - 4% Milestone 0 (Initial Planning)
    - 2% Milestone 1
    - 2% Milestone 2
    - 2% Milestone 3
    - 10% Final-Product Evaluation
  - 20% Individual Productivity
    - 5% Milestone 1 Regular Productivity
    - 5% Milestone 2 Regular Productivity
    - 5% Milestone 3 Regular Productivity
    - 5% Above and Beyond Productivity

To convert from percentages to letter grades, see the following table:

≥ 97%	91–96%	89–90%	87–88%	81–86%	79–80%	77–78%	71–76%	69–70%	67–68%	60–66%	≤ 59%
A+	A	A-	B+	B	B-	C+	C	C-	D+	D	F

I reserve the right to *lower* the percentage threshold for letter grades as I see fit (i.e., I may make the grading scale better for you, but never worse).

6.1   Skills Tests and Knowledge Tests
In lieu of taking a small number of long exams, students will instead be evaluated based on a large number of small tests. Most class sessions will begin with one of these tests.

- There will generally be a hard time limit of 20 minutes to complete a test.

- The tests will be administered in the same fashion as exams, and thus, will have strict rules. Violation of the rules will be considered cheating and will be prosecuted as such.

The tests will come in two flavors: *Skills Tests* and *Knowledge Tests*.

Skills Tests will have students perform actual coding tasks using their laptop software development environments. A key aim of the Skills Tests is to evaluate whether students can perform core development tasks quickly and correctly. Such skill and efficiency is essential to being an effective software development team member.

- It is the student's responsibility to have and maintain a working laptop development environment to use for the Skills Tests.
- Skills Tests will be graded only as pass/fail (i.e., no partial credit).
- During a Skills Test, students may search the web for documentation and existing help-forum posts; however, No communications, such as emailing, texting, posting, messaging, etc. will be allowed. Engaging in such communication will be considered cheating and will be prosecuted as such.
- To deter cheating, students will be required to record a screen-capture video of their performance of the test, and to submit the video along with any other submission materials. Without such a video, a test submission will be considered incomplete, and thus, the test will be considered failed.
- For each Skills Test, there will be a second chance to pass the test given one week after the test was first given. The second-chance test will generally be very similar to the first test, however, with some details changed.

Knowledge Tests will be paper-and-pencil tests that evaluate a student's understanding of key terms and concepts as well as their mastery of certain skills (i.e., ones amenable to paper-and-pencil testing).

- Knowledge Tests are cumulative in that the questions/problems may be drawn from any material covered prior to the test. It will not be uncommon for the same topic to appear on multiple tests.
- Knowledge Test questions will each have a weight, and individual questions will be graded only as pass/fail (i.e., no partial credit on questions).
- Knowledge Tests will generally be closed book, closed notes, closed neighbor.
- Unlike the Skills Tests, there will generally be no second-chance tests for the Knowledge Tests.

In general, makeup tests will NOT be administered if a student misses the test. If you have an extenuating circumstance, you should notify me as soon as possible. Makeups will only be given under extreme circumstances and if I approve the absence before the test is given. All excused absences must be documented (e.g., with a doctor's note).

6.2   Team Project

The centerpiece of this course is a team software project. Teams of roughly 4 students will work together to develop a software system for a customer.

I reserve the right to assign the teams, and to reshuffle them as I see fit.

Team projects in an educational setting must balance two concerns: (1) the need for students to work together as cohesive teams, and (2) the need for individual accountability. Thus, half of your project grade will be based on what your team is able to achieve as a whole and half will be based on your individual contributions to the project (i.e., your productivity).

6.2.1 Team Achievement

Teams will complete a series of project milestones, each with its own goals and instructions. Milestones will be evaluated based on criteria, which include the following:

- Quality of planning and design artifacts
- Effectiveness of communication
- Discipline in following software engineering processes and procedures

Additionally, at the end of the project there will be a final evaluation of the product produced by the team. This evaluation will focus mainly on the quantity and quality of features produced.

The marks awarded for each of the above items will generally applied to the team as a whole (i.e., everyone on the team will receive the same marks; see exceptions for low individual productivity below).

6.2.2 Individual Productivity

6.2.2.1 Regular Productivity

The majority of your individual productivity points are associated with *regular productivity*. Each team member will be assigned certain tasks for each milestone. It is expected that each team member will complete their assigned tasks in a timely manner. It is also expected that team members will be continuously productive, and not to put off their work, rushing to slap something together at the last minute. Failure to do so may result in deductions from your regular productivity grade.

6.2.2.2 Above and Beyond Productivity

To achieve the highest grades in the course (A/A+), you will need to go above and beyond the call of duty; thus, your individual productivity grade also accounts for *above and beyond productivity* (aka *A&B*). Throughout the course, you will have the opportunity to take on special roles or to negotiate A&B tasks to do in addition to your regular task assignments. There is no limit on the number of A&B points you can earn, but note that you will need at least 5 A&B points to get full credit.

6.2.2.3 Additional Productivity Policies

- **Deduction for Unproductiveness:** A student who demonstrates unsatisfactory productivity may also lose points on the associated milestone and also the final-product evaluation. Such deductions are meant to account for the lack of contribution made by an unproductive team member to the project.
- **Late Work:** You are expected to complete work on schedule, as deadlines are a part of the real world. Work will not be accepted late unless there are extenuating circumstances and prior arrangements are made with me.
- **Limit on weekly A&B earnings:** You may earn a maximum of 2 A&B points per week for negotiated A&B tasks. This policy is mainly to prevent students from putting off doing A&B work until the very end of the semester, and then flooding the instructor with low-quality work in an 11th-hour attempt to earn more points.

7 Accommodations for Disabilities

Reasonable and appropriate accommodations will be provided to students with disabilities who present a memo from Disability Resources for Students (http://www.memphis.edu/drs/).

8 Plagiarism/Cheating

Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and ***will not be tolerated***. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but ***appropriate references must be included for the materials consulted***, and appropriate citations made when the material is taken verbatim.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor's discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the Office of Student Conduct for further disciplinary action. For further information on U of M code of student conduct and academic discipline procedures, please refer to: http://www.memphis.edu/studentconduct/misconduct.htm.

8.1 Course-Specific Instructions

- Teammates (i.e., members of the same team) may collaborate and share work however they see fit; however, if asked to report what each team member's contributions were, students must provide honest responses.
- Students from different teams may not collaborate in this way.
- Teammate collaboration is limited to project work, and is not allowed on any other course work (e.g., homeworks, quizzes, exams), unless specifically noted.

8.2 TurnItIn (Academic Integrity)

Your written work may be submitted to Turnitin.com, or a similar electronic detection method, for an evaluation of the originality of your ideas and proper use and attribution of sources. As part of this process, you may be required to submit electronic as well as hard copies of your written work, or be given other instructions to follow. By taking this course, you agree that all assignments may undergo this review process and that the assignment may be included as a source document in Turnitin.com's restricted access database solely for the purpose of detecting plagiarism in such documents. Any assignment not submitted according to the procedures given by the instructor may be penalized or may not be accepted at all.