CpSc 3720: Introduction to Software Engineering
Spring 2016
Syllabus

Instructor: Stephen Schaub  
Office: McAdams 127  
Email: sschaub@clemson.edu  
Office Hours: TTh 1:00-1:50 p.m. 
others by appointment  
Lecture Hours: TTh 2:00-3:15 p.m., McAdams 119

Course Information

Prerequisites
CpSc 2120 and CpSc 2150. Significant experience with Java (from CpSc 2150) is expected.

Course Description
This course will provide an intensive introduction to software engineering. It will cover each major phase of the software lifecycle. It will provide introductory coverage of requirements analysis, object-oriented design, and project management, and intermediate coverage of module-level design principles, program specification and reasoning principles, and program validation and verification techniques. Oral and written communication issues and ethical issues are among the topics to be covered in the course.

Student Learning Outcomes
1. Graduates will be able to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
2. Graduates will be able to analyze a problem, and identify and define the computing requirements appropriate to its solution.
3. Graduates will be able to apply design and development principles in the construction of software systems that function in accordance with specifications.
4. Graduates will be able to function effectively on teams to accomplish a common goal.

Course Resources

Textbook Information
2. *Instructor Notes on Software Components*; available from the campus bookstore or online.

Course Website
Assignments and other course materials are available at [http://cs.clemson.edu/~sschaub/cpsc3720](http://cs.clemson.edu/~sschaub/cpsc3720)
Grading

<table>
<thead>
<tr>
<th>#</th>
<th>Item</th>
<th>Points</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Quizzes</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>Project Deliverables</td>
<td>50</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>Tests</td>
<td>75</td>
<td>150</td>
</tr>
<tr>
<td>1</td>
<td>Final Exam</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td><strong>Total Points:</strong></td>
<td></td>
<td><strong>500</strong></td>
</tr>
</tbody>
</table>

The instructor reserves the right to make adjustments to these grading calculations.

**Quizzes:** There will be regular quizzes, announced in class.

**Team Project:** A team project will give you an opportunity to practice the software engineering techniques discussed in the lecture. Each project deliverable will involve coordinating and integrating the work of each team member into a single team submission; the separate submission of individual deliverable components from students will not be accepted. Each student will be graded individually on the portions of the deliverable he worked on, as well as the quality of the overall team submission.

### Course Policies

#### Deadlines and Late Work

Assignments can receive full credit only if submitted in full by the beginning of class on the day due. Late submissions will not be accepted.

There will be no makeup quizzes or exams except for prior approval or valid medical excuse. The lowest quiz grade will be dropped at the end of the semester.

#### Attendance Policy

Students are expected to attend class and come to class on time. Students who miss class are responsible to find out what was covered. If the instructor has not arrived within 15 minutes of the scheduled start of class, you may leave.

For planned absences, students are expected to notify the instructor a week ahead of time. Written assignments and scheduled tests should be completed before a planned absence; contact the instructor to make arrangements for doing so. It is the student’s responsibility to check in advance of an absence to verify what is due.

For absences due to illness or emergency, contact the instructor via e-mail as soon as you realize you will not be in class in order to make arrangements for making up any graded work without penalty.
Accommodations for Students with Disabilities
If you have a documented learning disability or if you are impaired in some way (auditory, visual, cognitive, neurological, or physical), please notify the instructor within the first week of the course so that any necessary adjustments can be made.

Academic Honesty and Integrity Policy
It is expected that you will work ALONE on exams and quizzes. Evidence to the contrary will be regarded as academic dishonesty and will be dealt with according to the University policies on academic dishonesty. For details, please see the following: http://www.clemson.edu/academics/integrity/

Classroom Etiquette
I expect to foster a nurturing learning environment based upon communication and mutual respect in this class. Students are required to exhibit professionalism and respect for others in their deportment and behavior. Rudeness towards anyone in the classroom will not be tolerated.

To minimize distraction, all electronic devices are to be turned off when class begins. Your presence in class implies that you are prepared to listen attentively and to participate in class discussion and group activities.

Clemson University is committed to a policy of equal opportunity for all persons and does not discriminate on the basis of race, color, religion, sex, sexual orientation, gender, pregnancy, national origin, age, disability, veteran’s status, genetic information or protected activity (e.g., opposition to prohibited discrimination or participation in any complaint process, etc.) in employment, educational programs and activities, admissions and financial aid. This includes a prohibition against sexual harassment and sexual violence as mandated by Title IX of the Education Amendments of 1972.

Course Topics
- Development Models
- Unified Process
- Requirements Analysis
- Use Cases
- Prototyping and User Interface Design
- Project Management
- Object-Oriented Design
- Software Architecture
- Source Control
- Component Design
- Testing and Formal Verification
- Copyright and Patent Law