

School of Computing  
**CP SC 372: Introduction to Software Engineering**  
Spring 2011

Instructor: Murali Sitaraman  
Contact Information: McAdams 210; Phone: 656-6738; E-mail: [murali@cs.clemson.edu](mailto:murali@cs.clemson.edu).  
Web page: Listed under [www.cs.clemson.edu/~murali](http://www.cs.clemson.edu/~murali)  
Lecture Hours: TTh: 9:30AM – 10:45AM, Daniel 415  
Office Hours: MTW: 2:00PM – 3:00PM; other hours by appointment.  
Prerequisites: CP SC 212 and CP SC 215.

### Materials

1. *UML Distilled, Third Edition: A Brief Guide to the Standard Object Modeling Language*, M. Fowler and K. Scott, Addison-Wesley, 2002.
2. *Instructor Notes on Software Components*; will become available sometime in September.
3. Hand-outs.

### Course Description and Content

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, requirements modeling, design modeling, 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.

### Grading Policy

Performance in this course will be evaluated by class participation, exams, homework and team assignments, and quizzes. Requests for makeup exams are discouraged. NO MAKEUP EXAMS will be given without prior approval or valid medical emergency. Assignments are due beginning of the class when they are due, and *they need to be submitted in the classroom*. Later submissions will not get any credit at all. (Only selected parts may be graded for quizzes and assignments; the entire grade for the assignments will be based on those parts.)

Breakdown of points is given below:

Class participation	5%
Team Assignments and Presentations	30%
Quizzes	20%
Exam #1 (Date TBA)	20%

Exam #2 (Date TBA)	20%
Final Exam	5%

Letter grades will be assigned as shown below:

90% - 100%	A
80% - 89%	B
70% - 79%	C
60% - 69%	D
< 60%	F

### **Attendance Policy**

Attendance in this class is not mandatory, but you are responsible for all materials covered in classes. In addition, note that there will be a quiz (or a HW assignment) that is given (or due) in every class, each worth 1%. Two make-up chances will be given. Class participation is worth 1% for every three weeks of the course, for a total of 5% points.

### **Academic Integrity**

All exams and homework assignments are individual tasks, unless specifically designated as group tasks. 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 [http://www.cs.clemson.edu/html/academics/academic\\_integrity\\_2002.html](http://www.cs.clemson.edu/html/academics/academic_integrity_2002.html).

### **Learning and Feedback**

I expect to foster a nurturing learning environment based upon communication and mutual respect in this class. I will give serious consideration to any suggestion as to how to further such a positive and open environment.

I encourage you to give feedback on various aspects of the course, including but not limited to contents, assignments and exams, style, and treatment. I encourage active participation in the class. Your feedback is important for improving the quality of this course and that of undergraduate education in computer science, in general.

If you have a special need and feel that you need assistance with regard to lectures, reading assignments, or testing, please advise me of your needs as soon as possible.

**Detailed Syllabus** (subject to change; please see the web site for reading assignments)

Week #1: Course introduction and grading policies; Introduction to software life cycle.

Week #2: Software life cycle; Process models; Introduction to requirements analysis; Requirements assignment #1.

Week #3: Requirements analysis; Introduction to software design; Discussion of requirements assignment #1.

Week #4: Software design; Introduction to software components; Design assignment #2.

Week #5: Interface design issues; Issues in object-oriented reasoning.

Week #6: Discussion of assignment #2; Exam review; Exam;

Week #7: Component design and specification.

Week #8: Introduction to implementation design; Component assignment #3.

Week #9: Implementations; Extensions.

Week #10: Specification-based testing and tracing; Introduction to verification; Discussion of assignment #3.

Week #11: Analytical reasoning; exam review.

Week #12: Fall break; White box testing; Exam #2; Testing and verification assignment #4.

Week #13: Project management and metrics; Discussion of assignment #4.

Week #14: Additional reasoning examples.

Week #15: Software maintenance; course summary; Assignment #4 due. Thanksgiving break;

Week #16: Presentations.