

# CS 206 Testing and Verification Techniques in Software Engineering

## Winter 2009

<b>Instructor</b>	Gianfranco Ciardo	email: <a href="mailto:ciardo@cs.ucr.edu">ciardo@cs.ucr.edu</a>	<a href="http://www.cs.ucr.edu/~ciardo/">http://www.cs.ucr.edu/~ciardo/</a>
<b>Prerequisites</b>	CS 141 and CS 150 (or equivalent courses)		
<b>Units</b>	4		
<b>Office hours</b>	Tue 1:30pm-4:00pm	Room: EBUII 442	
<b>Class meetings</b>	Mon/Wed 2:10pm-3:30pm	Room: SPR 2344	(subject to change)
<b>Final examination</b>	Thu, March 19, 2009 8:00am-11:00am	Room: SPR 2344	

**Textbook** The course covers both software testing and software model checking, thus two books are required.

- Aditya P. Mathur, *Foundations of Software Testing: Fundamental Algorithms and Techniques*. 2008, Addison-Wesley Professional.
- Béatrice Berard et al., *Systems and Software Verification*. 2001, Springer-Verlag.

**Catalog description** Introduces techniques to verify that the software runtime behavior meets its specifications. Topics include model checking: safety, liveness, temporal logics, and abstraction; static and dynamic analysis: data flow analysis, concept analysis, program slicing, and invariant detection; testing: test generation, prioritization, suite reduction, and regression; automated debugging: fault location and visualization.

**Detailed contents** The following topics will be covered:

- Introduction and motivation
  - The need for software testing and verification
- Software testing:
  - Review of basics concepts of static and dynamic analysis
  - Unit testing and integration testing
  - Test generation techniques
  - Regression testing
  - Test selection and prioritization
  - Test adequacy criteria based upon control flow and data flow
- Model checking:
  - Review of basic concepts in logic and automata theory
  - State space generation
  - Linear time properties, safety vs. liveness properties, regular and  $\omega$ -regular properties
  - LTL and CTL model checking
  - Fairness
  - A software verification tool (SPIN) and its specification language (Promela)
- Advanced topics (coverage depends on students' interests and on available time)

**Coursework** The following coursework will be used to assess your final grade:

Homeworks (around 4):	50% of the grade.
Individual presentation of an advanced research paper	12.5% of the grade.
Final exam:	37.5% of the grade.

If you foresee having to turn in a homework after its due date for justifiable reasons (such as an illness), you *must* let me know by email or in person as soon as you can, and certainly before the deadline.

I *require* the use of LaTeX to typeset your homeworks, and I *strongly suggest* you use it for your presentations as well. If you are not familiar with LaTeX, you can find examples to get you started on my homepage, or on the web. Don't wait until your first homework is due!

**Cheating guidelines** Unless otherwise stated, all coursework is to be performed by you alone. If you find and use helpful information for an assignment (for example, on the web or in a book), you can incorporate it in your homework as long as you include a proper reference. In addition, verbatim citations should be clearly indicated as such. Finally, any programming work you turn in for this course should be your own. Anything else is considered cheating and will result in a zero score for the entire homework assignment, and probably much worse, depending on the gravity of the offense.

**Grading policy** Unless you successfully petition for a Satisfactory/No Credit (S/NC) grade, you will receive a letter grade, which will be determined as follows. Assuming your overall numerical grade is  $x\%$ , your letter grade is:

$-\infty < x < 57$	:	F	$57 \leq x < 60$	:	D-	$60 \leq x < 63$	:	D	$63 \leq x < 67$	:	D+
			$67 \leq x < 70$	:	C-	$70 \leq x < 73$	:	C	$73 \leq x < 77$	:	C+
			$77 \leq x < 80$	:	B-	$80 \leq x < 83$	:	B	$83 \leq x < 87$	:	B+
			$87 \leq x < 90$	:	A-	$90 \leq x < 93$	:	A	$93 \leq x < +\infty$	:	A+