



School of Computing

[cs.main](#) || [calendar](#) || [course.catalog](#) || [faculty](#) ||

Syllabus

CS 333

Software Engineering I

BYU

Hawaii

Fall 2005

Classroom:

GCB 178

Schedule:

Section 01:

MWF 2:00-2:50 PM (GCB 178)

Final Exam:

On or before TBA

Instructor:

Professor Randy Winters

Phone:

808-293-3471

Email: winters@cs.byuh.edu

Class Email: cs333@cs.byuh.edu

Office:

GCB 130A

Website:

<http://winters2.cs.byuh.edu/cs333.htm>

**Instructor's
Office Hours:**

MWF (By Appointment)

TR (By Appointment)

Updated office hours are posted outside my office door. Students for whom the posted hours are not convenient can contact me by email to make an appointment.

I also have an open-door policy. If my door is open (even just a bit) feel free to knock and come in. If I am not in my office during my posted office hours, I may be in the lab.

T.A. Hours: (TBA)

T.A. Room: (TBA)

Description: Software Engineering is the discipline concerned with the application of theory, knowledge, and practice for effectively and efficiently building software systems that satisfy the requirements of users and customers. Software Engineering is applicable to small, medium, and large-scale systems. It encompasses all phases of the life cycle of a software system. The life cycle includes requirements analysis and specification, design, construction, testing, and operation and maintenance.

Software engineering employs engineering methods, processes, techniques, and measurement. It benefits from the use of tools for managing software development; analyzing and modeling software artifacts; assessing and controlling quality; and for ensuring a disciplined, controlled approach to software evolution and reuse. Software development, which can involve an individual developer or a team of developers, requires choosing the tools, methods, and approaches that are most applicable for a given

development environment.

The elements of software engineering are applicable to the development of software in any computing application domain where professionalism, quality, schedule, and cost are important in producing a software system.

If you have a disability and would like accommodations, contact Ken Hurt (656-7516) in the Disability Resource Center (BRN, room 108).? The syllabus is available in alternative formats (large print, tape recorded, etc.).

Objectives: This course (CS 333) is the first of the two parts. The final course, CS 444 is offered next semester. Students will learn software engineering principles, practices and methodologies in this two part course.

Through classroom lecture and a software studio project (where students work in teams), students will:

- learn how to structure major software developments utilizing phases, stages and designs.
- learn about team and software process models including personal and team processes (PSP) and (TSP).
- learn about prescriptive process models such as the waterfall, incremental and evolutionary process models.
- learn about agile development and the use of agile models such as eXtreme Programming (XP).
- learn about software engineering practices in communication, planning, modeling, construction and deployment.
- learn about system modeling using UML and other processes.
- learn about requirements engineering and understand the basic requirements

- engineering tasks of inception, elicitation, elaboration, negotiation, specification, validation and requirements management.
- learn about analysis modeling to include data modeling, scenario-based modeling, flow oriented and class-based modeling.
 - learn basic design concepts such as abstraction, architecture, patterns, modularity, information hiding, functional independence, refinement, refactoring and design classes.
 - learn about formal software architecture and gain an understanding of the basic architectural styles such as pipe-and-filter, event-based, implicit invocation and layered systems to name a few.
 - learn about software testing strategies and testing techniques.
 - learn the basics of software project management including the management of the people, product and process components.
 - learn various processes for estimating and scheduling software projects.
 - learn about risk management and reactive vs. proactive risk strategies.
 - learn about quality and change management.

Prerequisites: CS 202 is a prerequisite. If you have not taken CS 202 but feel strongly that you have the necessary knowledge and background to qualify for entrance into this course please contact me and we can discuss this possibility.

Why Take This Course? The software industry is still in its early stages of growth and development. The developmental processes used today are still very ineffective and unpredictable.

The fact that three out of four large software projects will fail give you an indication of the

inadequacy of current software development processes. Software engineering brings tested and tried development methodologies and practices to the software development process.

Software engineering is a relatively new field which is producing highly paid professionals. The ability to understand, plan and manage the development process is a valuable asset to possess and will place you ahead of fellow programmers in the job market and will allow you to provide a successful development process to your employer. This translates to a higher paying job and an opportunity to start your career with knowledge and skills that will place you on a lucrative career path.

Attendance: Attendance is mandatory. I will take roll each day. Students are responsible for material covered and announcements made in class. School related absences may be made up only if prior arrangements are made.

Due to INS (immigration) and VA (veterans) requirements the Vice President for Student Life should be notified whenever a student misses four consecutive class days. In the context of this class, that happens when you fall behind and are not actively working on assignments for a period of ten calendar days.

Resources: There are three textbooks for this course. They will also be used for CS433 next semester.

1. **Software Engineering - A Practitioner's Approach?** (sixth edition) by Roger S. Pressman., McGraw Hill - ISBN 0-07-301933-X.

2. **"Extreme Programming Explained : Embrace Change"** (2nd Edition) by Kent Beck, Cynthia Andres Addison Wesley

3. **"Managing Software Requirements: A Unified Approach"** (The Addison-Wesley Object Technology Series) by Dean Leffingwell, Don Widrig
Addison Wesley

Reading: The student is responsible for reading the material in the textbook.? A reading schedule will be provided in the class schedule. The student is expected to read the material BEFORE the class in which it is discussed.

To get reading credit, you must let your sight rest on each of the words in the assignment, and you must try to understand what is being said. If you can speed-read some or all of it with reasonable comprehension, that is acceptable too.

Software Studio: A key component of this course is the actual development of a software project. In addition to class lecture time, students will be organized into teams of 4-6 students and will work on the development of a real software project which will span the duration of both the Fall and Winter semester (Software Engineering I & II) classes.

Notice: The class schedule presented is approximate.? The instructor reserves the right to modify the schedule according to class needs.? Changes will be announced in class and will be updated as necessary.

Tests: The test schedule will be announced in class. There will be no makeup tests given, except with **PRIOR** approval.?

Homework: TBA

Late Policy: Assignments are due on the date posted in the class schedule. Submitting them late will result in penalties.

Time Commitment: This class is the first of two classes (Software Engineering I & II). In order to benefit from the class, you must also take Software Engineering II in the Winters. The software teams formed in Software Engineering I, will carry over into Software Engineering II and the software project will be completed over both classes.

Software Engineering is usually presented over a two year period as a graduate-level class in many institutions. These two classes are intensive and pack a lot of material into two short semesters. It will be hard work requiring team programming and development outside the normal class lectures. The work will be hard. . . but worth the time you have invested. Your software engineering experience will give you a decided advantage over others as you enter the workplace.

It is estimated that you will need to spend about 6 hours per week outside of the regular class lectures for reading, assignments and for participation in the studio part of the class.

Cheating: Cheating will not be tolerated, and will result in a failing grade for the students involved.? Cheating includes, but is not limited to, turning in homework assignments that are not the student?s work.

You are encouraged to work in groups while studying for tests, discussing class lectures, discussing algorithms for homework solutions, and helping each other identify errors in your homework solutions.? However, each student must create and type in their own solution.? Any kind of copying and pasting is NOT acceptable.? If you need help understanding concepts, get it from me, the TA, or fellow classmates.? But never copy another?s code,

either electronically or visually. I consider any form of cheating an honor code violation.

Grading:

<i>Points</i>	<i>Grade</i>		<i>Points</i>	<i>Grade</i>		<i>Points</i>	<i>Grade</i>		<i>Percent</i>	<i>Grade</i>
930 +	A		830-869	B		730-769	C		630-669	D
900-929	A-		800-829	B-		700-729	C-		600-629	D-
870-899	B+		770-799	C+		670-699	D+		0-599	F

Incomplete & UW: If you quit working in the class, generally I will give you a "UW" grade. In addition to saying that you failed the class, a UW also tells people that you didn't seriously attempt the class; you just gave up.

I do not give "I" grades (incompletes) except in unusual circumstances. In my experience only a small fraction of incompletes are ever completed. I will consider giving you an incomplete if you request it, seem to have a good reason, have a pretty solid timeline for completion, and you get the necessary paperwork filled out. I will not babysit you through the completion of an

Incomplete, but I will assist you when you ask for help.

Course Calendar: Please see the class schedule.

Special Needs: Brigham Young University Hawaii is committed to providing a working and learning atmosphere, which reasonably accommodates qualified persons with disabilities. If you have any disability that may impair your ability to complete this course successfully, please contact the students with Special Need Coordinator, Leilani A'una at 293-3518. Reasonable academic accommodations are reviewed for all students who have qualified documented disabilities. If you need assistance or if you feel you have been unlawfully discriminated against on the basis of disability, you may seek resolution through established grievance policy and procedures. You should contact the Human Resource Services at 780-8875.

Preventing Sexual Harassment: Title IX of the education amendments of 1972 prohibits sex discrimination against any participant in an educational program or activity that receives federal funds, including Federal loans and grants. Title IX also covers student-to-student sexual harassment. If you encounter unlawful sexual harassment or gender-based discrimination, please contact the Human Resource Services at 780-8875 (24 hours).



Brigham Young University - Hawaii

55-220 Kulanui Street - Box 1854 - Laie, HI
96762-1294

Copyright (c) 2004 - BYU Hawaii

Retrieved from: <http://winters2.cs.byuh.edu/Syllabus-CS333.htm>