

Master Course Syllabus
School of Engineering and Computer Science
Washington State University Vancouver

CS 320
Fundamentals of Software Engineering
3 Semester Hours
(3 lecture hours)

Catalog Description

Introduction to software engineering; requirements analysis, definition and specification; software process models; prototyping; architecture; object-oriented design with UML.

Prerequisite Courses

CS 224	Programming Tools
Math 216	Discrete Structures
Engl 402	Technical Writing (concurrent enrollment allowed)

Prerequisite Topics

- Experience with an Object-Oriented Programming Language (Java or C++)
- Principles of technical writing
- Use of UNIX or Windows environment for coding, compilation, debugging and testing

Measured Course Outcomes

Students taking this course will:

1. Elicit and analyze requirements to create an internally consistent requirements specification. B-1
2. Verify and validate the requirements specification over the course of a software project's lifecycle. B-2
3. Analyze various software lifecycle processes and their relation to cost, timeliness, quality and changing requirements. C-1
4. Use a specific software lifecycle model to develop a design that is consistent with specified requirements. C-2
5. Use a software lifecycle model to implement a design, satisfying specified requirements. C-3
6. Create a test suite for unit testing, integration testing, acceptance testing or regression testing and report test results. C-4

Required Textbooks

Software Engineering, A Practitioner's Approach, Roger Pressman,
Mc Graw Hill, 7th Edition

Reference Material

Software Engineering Theory and Practice (3rd Edition), Shari Pfleeger
and Joanne Atlee, Prentice Hall.

UML Distilled (Third Edition), Martin Fowler, Addison Wesley Inc.

The Mythical Man-Month (Anniversary Edition), Frederick P. Brooks,
Addison Wesley Inc.

Major Topics Covered in the Course

1. Object Oriented Design and Programming
2. Software Requirements and Specification
3. Software Engineering Processes
4. Testing and Debugging
5. Elements of Effective Oral Communication

Laboratory Projects

Programming Project Area	Weeks
Object oriented programming	4
Object oriented design using a graphical UML tool	2

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	0	Computer Organization and Architecture	0	0
Algorithm & Software Design	0	2	Concepts of Programming Languages	0	1

Oral and Written Communications

This is a "writing in the major" course, thus a minimum of two papers are required. This first writing assignment is a 2500 - 3000 word report on a

specific software methodology. The report includes a critique of several aspects in the methodology. The second writing assignment is part of a semester long project. The student must write a software-requirements-specification (SRS), which is a technical document that describes the requirements of a software project. Students must keep a hand written project journal that discusses the progress and problems of their project. In addition, students will present one oral presentation covering their semester long project. The oral presentation will be used to critique their presentation skills as well as asses their semester long project.

Social and Ethical Issues

This course discusses either IEEE's or the ACM's code of ethics.

Theoretical Content

Topic	Hours
Formal methods	3
Test Coverage	3

Problem Analysis

The instructor performs analysis of representative problems in class. All student programming assignments require the student to analyze problem requirements. The instructor analyzes problem solutions (both his own and student's) in class.

Solution Design

This course introduces students to software engineering with emphasis on requirements analysis, specification, object-oriented design, implementation and testing (unit and integration). Students incrementally develop a software product to give them hands-on experience with software engineering principles. There will be documentation (including the appropriate artifacts) required as entrance and exit criteria to each phase of the process. The objective is to experience particular methods in each phase of the software development lifecycle.

CC2001

This course provides coverage of topics in the following areas (hours listed are minimums):

SE1. Software Design [core]	8
SE2. Using APIs [core]	2
SE3. Software Tools and Environment [core]	3
SE4. Software Processes [core]	2
SE5. Software Requirements and Specification [core]	4

SP4. Professional and Ethical Responsibilities [core]	1
---	---

Course Coordinator: Orest Pilskalns
Last Updated: Feb 1, 2008 (Approved)
Syllabus Version Number: 1.3