

## COMPSCI 471/571: Software Engineering

### Instructor

Instructor: Jim Buffenbarger  
Electronic mail: [buff@cs.BoiseState.edu](mailto:buff@cs.BoiseState.edu)  
Office: MEC-302C 426-3567

### Meetings

Lectures: TuTh 12:00-1:15 MEC-309  
Office hours: TuTh 1:30-2:30 MEC-302C  
by appointment MEC-302C

### Catalog Description

A formal study of the software development process. Topics include: lifecycle models, requirements definition, specification, design, implementation, validation, verification, maintenance, and reuse. Students work in small teams on significant projects.

### Objectives

Students are introduced to basic concepts of software engineering and software development, including:

- software lifecycles and the need for a rigorous approach to software development
- software build-management and configuration-management techniques and tools
- informal but systematic methods of documenting the requirements and design of a software system (e.g., UML)
- software-design methods, including procedure-oriented and object-oriented design
- formal methods of specifying software
- the experience of working as a member of a team during the early phases of a development project
- other contemporary topics

## Prerequisites

COMPSCI 125 Introduction to Computer Science I  
COMPSCI 225 Introduction to Computer Science II  
COMPSCI 342 Data Structures and Algorithms

## Textbook and Other Resources

The textbook is:

- *Object-Oriented Software Engineering*, by Stephen Schach. First edition. McGraw-Hill, 2008, ISBN-10: 007352333X, ISBN-13: 9780073523330.

## Activities

Grades are based on student performance of several kinds of activities. Their weights are listed below. If you organize and manage your time effectively, the workload should not exceed 10–15 hours per week, including class meetings.

|          |     |
|----------|-----|
| Homework | 20% |
| Exam     | 25% |
| Project  | 30% |
| Final    | 25% |

## Project

Students form and work in small development teams to produce a Specification Document for a realistic product comprising both hardware and software. Each team is expected to consult with, and submit progress reports, to the instructor, who acts as their customer. Documents are typeset using publishing software, nominally Latex and Dia.

Later in the semester, homework assignments ask each student to develop formal design specifications for interesting parts of his or her team's project.

## Exams

Two exams are administered during the semester. They are in-class, open-note, and open-book tests.

## Grading

Homework is delivered at the beginning of class on the day it is due. Late work is not accepted.

Makeup examinations are not normally administered.

Scores are posted near my office, as they become available. You are encouraged to check your scores to ensure they are recorded properly. If you feel that a grading mistake has been made, contact me within two weeks of the date that work is returned. Old scores are not changed.

## Academic Honesty

The following quotation is from the BSU Undergraduate Catalog. You should read that section.

*The university's goal is to foster an intellectual atmosphere that produces educated, literate people. Because cheating and plagiarism are at odds with that goal, they shall not be tolerated in any form. Therefore, all work submitted by a student must represent that student's own ideas and effort; when the work does not, the student has engaged in academic dishonesty.*

There is related material in the BSU Student Handbook.

*The course instructor is responsible for handling each case of academic dishonesty in the classroom except where a major or repeated offense is involved. In a proven case of cheating a student will be dismissed from the class and a failing grade issued.*

There are many forms of academic dishonesty. Some relevant examples include:

- Submitting programs, or parts of programs, written by someone else.
- Viewing exam answers, homework answers, or programs written by someone else. This includes material from other courses and previous semesters.
- Distributing exam answers, homework answers, or programs to someone else, even after it has been graded.

The BSU Undergraduate Catalog contains more examples. If you are unsure about a particular case, ask your instructor,

On homework, a student must work independently. Ideas and general principles can be discussed with other students, but work must be original.

Keep your files to yourself. See the UNIX commands `chmod go-rwx` and `ls -l`.

On projects, students work in teams. Indeed, the experience of working on a development team is an important aspect of this class. Of course, each member must contribute to the project. If you experience managerial problems with members of your team, which you cannot resolve internally, let me know early! If you ignore such problems early in the semester, they can easily undermine your team's ability to finish its project.

On exams, of course, each student must work entirely independently.

## **Computer Accounts**

Each student receives an account on the department's network of computers, which run the LINUX operating system. If you are unfamiliar with the department's computers, you are urged to attend office hours during the first week or two of classes. I'll try to get you started with these powerful tools.

You are responsible for understanding and complying with the departmental computing policy.

## Schedule

| <i>Week</i> | <i>Date</i> | <i>Topic</i>                      | <i>Assigned</i> | <i>Due</i> | <i>Reading</i> |
|-------------|-------------|-----------------------------------|-----------------|------------|----------------|
| 1           | Aug 28 Tue  | The Scope of Software Engineering |                 |            | 1              |
|             | Aug 30 Thu  |                                   |                 |            |                |
| 2           | Sep 04 Tue  | Software Lifecycle Models         |                 |            | 2              |
|             | Sep 06 Thu  |                                   |                 |            |                |
| 3           | Sep 11 Tue  |                                   |                 |            |                |
|             | Sep 13 Thu  |                                   |                 |            |                |
| 4           | Sep 18 Tue  | The Software Process              | HW#2            |            | 3              |
|             | Sep 20 Thu  |                                   |                 |            |                |
| 5           | Sep 25 Tue  | Teams                             |                 | HW#2       | 4              |
|             | Sep 27 Thu  |                                   |                 |            |                |
| 6           | Oct 02 Tue  | The Tools of the Trade            | project         |            | 5              |
|             | Oct 04 Thu  |                                   |                 |            |                |
| 7           | Oct 09 Tue  | Testing                           |                 |            | 6              |
|             | Oct 11 Thu  |                                   |                 |            |                |
| 8           | Oct 16 Tue  | From Modules to Objects           | HW#3            |            | 7              |
|             | Oct 18 Thu  |                                   |                 |            |                |
| 9           | Oct 23 Tue  | Reusability and Portability       |                 |            | 8              |
|             | Oct 25 Thu  |                                   |                 |            |                |
| 10          | Oct 30 Tue  | Requirements                      |                 | HW#3       | 10             |
|             | Nov 01 Thu  |                                   |                 |            |                |
| 11          | Nov 06 Tue  | Classical Analysis                | HW#4            |            | 11             |
|             | Nov 08 Thu  |                                   |                 |            |                |
| 12          | Nov 13 Tue  | Object-Oriented Analysis          |                 |            | 12             |
|             | Nov 15 Thu  |                                   |                 |            |                |
| 13          | Nov 20 Tue  | Thanksgiving                      |                 |            |                |
|             | Nov 22 Thu  |                                   |                 |            |                |
| 14          | Nov 27 Tue  | Design                            |                 |            | 13             |
|             | Nov 29 Thu  |                                   |                 |            |                |
| 15          | Dec 04 Tue  |                                   |                 | HW#5       |                |
|             | Dec 06 Thu  |                                   |                 |            |                |
| 16          | Dec 11 Tue  |                                   |                 | project    |                |
|             | Dec 13 Thu  |                                   |                 |            |                |
| 17          | Dec 20 Thu  | Final: 12:00-2:00                 |                 |            |                |