

---

## INTRODUCTION

---

This one hour course is designed as a co-requisite with any of the computer science department's programming language courses. Currently, these are CS137, Pascal; CS157, Fortran; CS167 C; CS177, C++. Java is being offered starting in Fall 1997. There are two aims for this course. The first is to improve each of the language courses by giving students important hands-on experience with basic principles of software engineering. Specifically, this means principles of code reading, program design, testing and debugging. The second aim is to free each of the courses from the necessity of teaching these principles. Students who take more than one language course should not have to cover the same programming methodology again. It will be offered as a co-requisite every Fall and Spring semester.

### WHO SHOULD TAKE THIS COURSE?

CS117 is a co-requisite for CS137, CS157, CS167 and CS177. Students who have not taken any language must take it concurrently with the language course. Students who have taken it need not take it again even though they may choose to take another language course. Typically, a student will take CS117 and CS167 in one semester, and CS177 later. Waivers are available to those students who can demonstrate experience in writing medium to large scale programs, or who have taken part in a group programming project.

### WHY YOU SHOULD TAKE THIS COURSE?

The legalistic answer is that you cannot take a language course without taking CS117 once. The more important educational answer is that learning to program is not just about learning the language. It is about problem-solving, organization and communication. This course will give you enough skills to be a good programmer, not just someone who knows the language, whether it is Pascal, C, Fortran, C++, or any other.

### WHY NOT TAKE SOFTWARE ENGINEERING?

The computer science department offers a course for CS majors called Software Development (CS371). In that course, students learn how to work on a large software project in small groups. This is the most important method of program development in industry. CS majors need to have a taste of this before tackling the real thing at their first job. However, many people are *occasional* programmers; they are not professional programmers, but need to write a program to carry out some special task in their own field. They might be biologists writing a simulation, engineers writing a control program, or almost anyone doing some form of data analysis. They very often work alone, so a full-blown course in software engineering is not appropriate. Moreover, there are some basic principles of programming that our majors acquire during their 200 level courses that are assumed in CS371. It is these basic principles that are the subject of this course. Having completed this course, you will probably be ready for CS371, but in this case, you probably ought to be a CS major anyway.

### WHERE'S THE TEXTBOOK?

We have yet to find a book that fills in the gap between the many excellent textbooks on individual languages, and the introductory software engineering textbooks. These class notes are intended as supplemental to your language textbook, just as the course is supplemental to the learning of the language. The material is language-independent. That is, the principles covered in the course can be applied to any or all of the languages taught in the department. As new languages are added, such as Java, this course will be adapted to adequately target this new developments.

## OVERVIEW

The course is in five parts. They are:

**code-reading:** this concerns syntax, program structure and layout

**hand simulation:** the basis of all testing and debugging methods

**design:** this covers both top-down and object-oriented methods

**testing:** how to tell whether your program is working right

**debugging:** how to track down bugs, using tracing, assertions, exceptions, and debugging software

## ASSIGNMENTS

There are no assignments in this course *per se*. However, for each of the five modules your instructor in the language course will give you an assign that will incorporate the ideas from the module. These assignments will, of course, be specific to the language you are learning. They will form part of the set of assignments for the language course. Your grade in this course will be awarded according to your performance in these five assignments. The instructor for this course will work closely with the instructors in the language course to make sure that continuity is maintained.

It is the aim of the language courses to reach a good level of knowledge by week five of the semester, so that the principles learned here can be used in the subsequent assignments. However, everything cannot be learned at once, so in this course we start off with the only fully language-independent module - design. By week five, sufficient knowledge will have been acquired to successfully tackle the code-reading and hand simulation assignments. The breakdown of the module assignments, by week is thus:

Week 2: design

Week 5: code reading

Week 6: hand simulation

Week 8: testing

Week 11: debugging