
\$Id: syllabus-cmps112.mm,v 1.83 2015-09-22 14:09:22-07 - - \$

PWD: /afs/cats.ucsc.edu/courses/cmps112-wm/Syllabus

URL: <http://www2.ucsc.edu/courses/cmps112-wm/:/Syllabus/>

1. General Information

The generic part of the syllabus contains detailed information about prohibiting cheating, due dates and times, submitting assignments, and verification of the submit. Read it carefully, as you will be held responsible for it.

Directory: The directory `/afs/cats.ucsc.edu/courses/cmps112-wm/` and its subdirectories contain all assignments, handouts, examples, old exams, etc.

Piazza: <https://piazza.com/> is for questions and discussions that are appropriate in the classroom or lab section.

Assignments: All assignments must be submitted electronically and must work on the IC Linux servers (`unix.ic`) in order to receive a grade. Submit programs using the submit command:

```
submit cmps112-wm.f15 ...
```

Due Dates: Due dates are announced in the **README** files in the course directory and in the newsgroup. You must frequently check the **README**. Late assignments are *not* permitted, except in emergency situations.

Cheating: ***Cheating will not be tolerated. See the section on cheating in the generic part of the syllabus.***

Textbook: Kenneth Louden, Kenneth Lambert: ***Programming Languages: Principles and Practice***, 3rd ed. Cengage Learning, 2012. ISBN-0-534-95341-7. The 2nd edition of this book is also still useable and probably much cheaper. Anyway, my notes are made from the 2nd, not the 3rd, edition. Priority level is “optional”. You will likely be able to get as much relevant information from good use of Google.

Grades: In order to pass the course, both the programming component and the testing component will be taken into consideration. Failing either component may be cause to fail the course. Your final grade and narrative evaluation will be based on the following allocation of points:

Programming assignments:	$4 \times 12\% = 48\%$
Midterm tests in class:	$2 \times 13\% = 26\%$
Final exam during exam week:	26%

2. Course Description from Catalog

CMPS-112. Comparative Programming Languages. Covers several programming languages and compares styles, philosophy, and design principles. Principles underlying declarative, functional, and object-oriented programming styles are studied. Students write programs emphasizing each of these techniques. Prerequisite: CMPS-101 or CMPS-109.

3. Syllabus

The course will follow two parallel tracks, each occupying about half of the lecture time. One track will consist of programming language principles and paradigms, and the other will detail some specific programming languages.

Principles. The following topics will be covered, with examples taken from various programming languages.

- Language design principles.
- Syntax and semantics.
- Data types.
- Expressions, statements, and procedures.
- Abstract data types and modules.
- Object-oriented programming.
- Functional programming.
- Logic programming.
- Parallel programming.

Practice. There will be four programming assignments, each in a different language, each showing a different programming paradigm :

- A scripting language.
- A dynamically typed functional language.
- A statically typed functional language.
- A dynamically typed object oriented language.

4. Generic Syllabus

Also read the generic syllabus in the directory `generic-syllabus/`.

5. Pair Programming

You may do pair programming if you choose. You are responsible for choosing a partner with whom you can work. Read the guidelines in the directory `pair-programming/`.

6. Submit checklist

Carefully read the submit checklist in `submit-checklist/` and make sure you understand how to use `submit`. All programs will be graded on the files you submit before the due date. Programs which work in your directory but not in the submit directory are not relevant to anything and will not be considered.

Forgetting to submit even a single file or submitting the wrong version of a file will have a very significant negative impact on your grade.