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/cse112-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 by using the **submit** command when logged into one of the Linux servers (**unix.ucsc.edu**). Assignments submitted via email will not be accepted. Files copied into the submit directory by any means other than the **submit** command (e.g., via **mkdir**, **cp -R**, **FileZilla**, etc., will be deleted and not accepted.

Due Dates: Due dates are announced in the **README** files in the course directory. You must frequently check the **README**. Emailed programs will not be accepted. *** Late submissions or makeups will not be accepted except in case of emergency (illness or injury) requiring a physician's attention.***

Cheating: Cheating will not be tolerated. See the secion on cheating in the generic part of the syllabus. Read the UCSC Academic Misconduct Policy for Undergraduates:
https://ue.ucsc.edu/academic-misconduct.html

Grades: The final grade will be determined by the following:

Programming assignments:	52% (4 programs × 13% each)
Midterm exam:	21%
Final exam:	27% (two hours)

To pass the course, it is necessary to pass both the programming and examination components of the course.

2. Course Description from Catalog

CSE-112. Comparative Programming Languages. Covers several programming languages and compares styles, philosophy, and design principles. Principles underlying functional, object-oriented, declarative, and scripting programming styles are studied. Students write programs emphasizing each of these techniques. Prior experience with Unix is assumed. Prerequisite: CSE-101 or CSE-111.

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.

3.1 Syllabus — 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.

3.2 Syllabus — Practice.

There will be four programming assignments, each in a different language, each showing a different programming paradigm:
- A dynamically typed functional language.
- A statically typed functional language.
- A dynamically typed object oriented language.
- A scripting language.
- A logic language.

4. Textbook

No textbook is explicitly listed for this course. Tutorials for the various languages we will be discussing can be found for free on the web.

`/afs/cats.ucsc.edu/courses/cse112-wm/Languages/`
`/afs/cats.ucsc.edu/courses/cse112-wm/Lecture-notes/`
`https://www.google.com/`

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/`. Be sure to submit a `PARTNER` file as per the specifications.

6. Submit checklist

Also read the submit checklilst in `submit-checklist/`.

7. Generic Syllabus

`$Id: generic-syllabus.mm.so,v 1.7 2019-08-16 14:22:43-07 - - $`

This is an attachment that I use for the syllabus in all of my courses. Any contradictions between what follows and what is written in the course-specific syllabus is resolved in favor of the course-specific syllabus.

8. Disability Accommodations

It is UCSC policy to assist students with disabilities. If you qualify for accommodations because of a disability, please submit to me your *Accommodation Authorization Form* from the Disability Resource Center (DRC) right away. Contact DRC at:

Disability Resource Center	Email: drc@ucsc.edu
146 Hahn Student Services	Phone: (831) 459-2089
1156 High Street	Fax: (831) 459-5064
University of California	
Santa Cruz, CA 95064-1077	

Student responsibilities are as follows:

(1) Students contact the DRC to determine their eligibility for accommodations. When approved by DRC, they will receive their Accommodation Authorization form.

(2) Students then notify their instructor during office hours or after class of their accommodations, and provide their instructor with their Accommodation Authorization Letter. This should be done during the *first week* of the quarter.

(3) Please note that it is the student's responsibility to contact the instructor about authorized accommodations. For disability-related testing accommodations, arrangements must be made at least three weeks before each test or exam.

9. Academic Integrity and Cheating

Cheating will not be tolerated. Cheating is defined as giving or receiving unpermitted aid in any programming assignments, examinations, or other course work that is used by the instructor as a basis of assigning grades. Incidents of cheating will be reported to the Provost of the student's college and to the School of Engineering for disciplinary action. Cheating in any part of the course may lead to failing the course and suspension or dismissal from the University.

Be warned that the School of Engineering has a policy of being highly intolerant toward cheating and/or academic dishonesty. All students shall read the UC Santa Cruz Academic Integrity web pages: `http://www.ucsc.edu/academics/academic_integrity/`.

Students are expected to maintain high standards of academic honesty. That means that any work submitted by a student which is not completely his/her own is not acceptable. The only exceptions are: code provided by the instructor or TA, whether given in class or provided in the instructor's course directory, and code

taken from the assigned textbook. Specifically, what is not acceptable is swapping code. "Just working together" is not an acceptable excuse. Helping each other with general questions is OK and that is one of the uses of the newsgroup.

Pair programming: For those courses where pair programming is explicitly permitted, it is not cheating for the two partners of a pair to share code, they are developing their projects using a single code base. See the guidelines for pair programming in another directory. It is cheating if the code between different pairs is excessively similar.

Any code not written by the student must be acknowledged in the `README` submitted with the assignment. Submitting code not written personally by the student and which is not acknowledged in the `README` is always cheating, whether or not the code would be otherwise authorized. Submitting code not written personally by the student, even if acknowledged in the `README`, is cheating if it pertains to parts of assignments that the student is expected to write individually. This refers to programs received from anyone, even tutors, or found on the web or other open-source code sources. And if an argument toward honesty does not convince you, note that if you can surf the web to find some code, so can others too, and they may then submit the same code you do.

Getting help from tutors in developing programs is of course expected. Tutors, in the course of their duties, may provide small code fragments, or explain to students how to rewrite their code, or how to debug existing code and modify it in order to get it working.

10. The Final Exam

From the Registrar: ***Final Examination Policies and Schedule.*** Final examinations are given during the exam week period at the time announced in the Schedule of Classes, usually in the same room used for class meetings during the quarter. Final examinations are required in all undergraduate courses. ... If a student misses an examination due to a documented illness or other emergency, the instructor may agree to give an Incomplete and schedule a makeup examination provided that the student's work is passing up to that point. When a final examination is one of the regular requirements in a course, no one taking the course may be individually exempted from it. Travel plans for vacation are not an emergency, and should not be made without checking the final examination schedule.

11. Assignment Specifications and the Final Grade

Late assignments will not be accepted except under extremely unusual circumstances and then only with adequate formal documentation. Make-up tests and exams will not be given, and workarounds for missed tests and exams will be permitted only in case of emergency, such as illness or hospitalization requiring the attendance of a physician.

A final grade will in general be given by the standard weighted average formula, where the values of $weight_i$ are given under 'Grades' at the beginning of the syllabus:

$$finalscore = \sum_{i=1}^{n}\left(\frac{score_i}{maximum_i} \times weight_i\right) \qquad \textbf{where} \qquad \sum_{i=1}^{n} weight_i = 1.0$$

The `Assignments` subdirectory contains the syllabus and assignments, given in both text and Postscript format. Both are generated from the same `groff` markup source file, but due to the restricted nature of ASCII text, some information may occasionally be missing, such as diagrams or math formulae. An example is the formula above.

If you are reading the ASCII text version of this file, you will see `geqn` markup language, while if you view the Postscript version, you will see the actual equation. The Postscript format version of the assignment is "official", and the only version that has been proofread. In case of discrepancy, refer to the Postscript version of the assignment. The text version is convenient for `grep`ping, but the final word is always in the Postscript version.

12. Due dates and times

A `unix.ucsc.edu` account is required in order to submit assignments. All assignments will be submitted electronically via the Linux timeshares and must work on these machines in order to receive a grade. Submitting an assignment via email is not acceptable.

Graders are required to be logged into the ITS Learning Technologies Linux servers (`unix.ucsc.edu`, etc.), machines when doing any part of the grading. Whether or not an assignment "works" with another operating environment will not influence an assignment score in any way. Students may develop their work with any operating environment they have access to, but must port their programs to ITS LT Linux before submitting them.

Exactly when are assignments due? Assignments are to be submitted electronically using the submit command any time on or before the date specified as the due date. That includes up to and including 23:59:59 of the specified due date. The time of the lecture is not relevant to the time an assignment is due. In practice, however, you can get away with submitting assignments even later, up until the time that the submit directory is locked with a Unix command. This will done sometime the morning after the listed due date.

Waiting until the last minute to submit is not a good idea, since after midnight is gremlin time. Gremlins are heavily armed with very sharp pointers. This leaves the exact cutoff time somewhat fuzzy. If you submit the same assignment more than once, the later submission will override the earlier one. ***Submit early! Submit frequently!***

Not knowing how to use the `submit` command is not an excuse for failing to submit an assignment. It is your responsibility to learn how to use it as well as how to use AFS well in advance of the due date. Once you have done the submit, verify that you have submitted everything that needs to be submitted.

13. Submitting assignments

Firstly, it is important for you to understand how to submit an assignment. Waiting until the due date to find that out will likely mean that you will not be able to submit, and hence will score zero.

Submit homework with the command :

```
submit volume project files ...
```

The volume name is the registrar's catalog code, a hyphen, the instructor's initials, a period, the quarter, and the two-digit year. Using `submit` without operands displays the possible submit volumes. If you specify a submit volume without a project name, the available project names will be displayed.

Finally, in order to verify that your submit is complete, and that the grader will be able to do the build, create a new directory in your personal file space, copy all of the files you have submitted and then perform your own test build.