Course Outline

- Introduction to several major high-level programming languages and their characteristics.
- This semester we will study four languages: Scheme, Haskell, Prolog, and Icon.
- At the end of the course you should be familiar with functional, logic, and string programming language paradigms.
- You should be competent programmers in Scheme, Haskell, Prolog, and Icon.
- You should no longer be daunted by the prospect of having to learn new languages and formalisms.
Late policy

- Assignments handed in no more than 24 hours late will incur a 5% penalty.
- Assignments handed in more than 24 but no more than 48 hours late will incur a 10% penalty.
- Assignments handed more than 48 hours after the deadline will receive a grade of 0.

Grading...

- I reserve the right to scale final grades.
- Scaling can be both “up” and “down”, depending, for example, on whether the final exam turned out to be particularly easy or hard.

You cannot make up an in-class test unless
1. you have notified us in writing (email is fine) or by phone prior to the test that you will be absent, and
2. you can produce a note from your doctor saying that you were incapable to take the test, and
3. you receive permission from the instructor prior to the test.

Description of Course

- Introduction to several major high-level programming languages and their characteristics.
- Programming projects are required in at least three languages.

This semester we will study four languages: Scheme, Haskell, Prolog, and Icon.

The official course description is at http://garnet.ccit.arizona.edu/schedule.cgi?CxSCz372z044zOpen

Honors Section

Depending on the interests of the students, we may
- Read and present research papers on programming languages.
- Learn and present programming languages not covered in class.
- Get involved with a research project.
**Course Objectives**

- At the end of the course you should be familiar with functional, logic, and string programming language paradigms;
- You should be competent programmers in Scheme, Haskell, Prolog, and Icon.
- You should no longer be daunted by the prospect of having to learn new languages and formalisms.

**Required Texts**

- Clocksin-Mellish, Programming in Prolog [optional].
- Various web resources.
- Lecture notes.

**Attendance Policy**

- You are not required to attend lectures, but... **you cut class at your own risk.**
- Anything covered in class or in any of the required readings is fair game on tests and exams.
- The honors section has mandatory attendance. We will probably not meet every week, but when we do meet, you have to be there.

**Subject to Change Policy**

- We may add, drop, or change topics.
- We may change exam or homework dates, etc.
- Changes will be announced in class and on the class web site! You are responsible for checking this site regularly.
- You should also check the course news group cs.course372 for announcements.
Notification of Objectionable Materials

- Some of the material may be hard, boring, or both.
- The instructor is known to sometimes make jokes in class which
  a) are not funny, or
  b) may be slightly off-color.  
  He apologizes in advance.
- Assignments and examples may touch on subjects which some may deem questionable, such as sex, war, and rock’n’roll.

Computer Access/Setup

You will be completing your homework on the department’s instructional machine, Lectura. You will also have access to the department’s lab in Gould-Simpson 228. You can access Lectura over the network or by dialing in. You will, therefore, need to set up an account on Lectura. To do so, go to the seventh floor of Gould-Simpson during normal business hours during the first few days of the semester and follow the instructions for setting up an account. When you apply for your account, you will pick up an application form. Fill out and return the form to Gould-Simpson 721 to pick up a magnetic access card that will allow you 24-hour access to the Gould-Simpson 228 lab.

Handicapped Accessibility

Students with disabilities who require reasonable accommodations to fully participate in course activities or meet course requirements must register with the Disability Resource Center. If you qualify for services through DRC, bring your letter of accommodations to me as soon as possible. See http://drc.arizona.edu.

Student Code of Academic Integrity

- Assignments in this course require individual attention and effort to be of any benefit. All work is expected to be that of each student alone. You may not consult with others, except in ways specifically authorized by the course instructor. You also may not plagiarize another person’s work or copy another person’s code.
Student Code of Academic Integrity...

- Code of Academic Integrity. Students are responsible for understanding and complying with the University’s Code of Academic Integrity. A synopsis of the Code is attached; the full text is available from the Office of the Dean of Students in Room 203 Old Main. Among other provisions, the Code demands that the work you submit is your own, and that graded papers and exams will not subsequently be tampered with. Copying of another student’s programs or data, or writings is prohibited when they are part of a published class assignment; it is immaterial whether the copying is by computer, xerox, pen or other means. Witting collaboration in allowing such copying is also a Code violation.

- Assignments in this course require individual attention and effort.
- Violations of the Code will, at minimum, result in loss of credit for a graded item. An egregious first violation or any second violation will minimally result in failure of the entire course.
- See also http://info-center.ccit.arizona.edu/~studpubs/policies/cacaint.htm the University of Arizona Code of Academic Integrity.

I take academic integrity seriously! I will report every violation!

Course Methodology

- I will lecture, you will learn.
- It is important in this class to allot significant time outside of class to programming in the new languages we will study.
- You cannot pass this class by cramming before the final. You must be a confident programmer in each language to pass the tests and exam, and such skills can only be acquired by daily hands-on practice.

- Most of the languages we will study have free implementations. If you own your own computer it’s a good idea to download and install the interpreters so that you can work at home.
- Just doing the programming assignments may not be enough to become a good programmer in the new languages. You should spend time outside class working programming exercises.
Assignment Format

- Assignments will be mostly in the form of programming problems.
- You may work the assignments on any machine you want, but before you hand them in, you should test the code on lectura! The TA will grade the assignments on lectura, and if they don’t work there, he won’t debug them for you! There can be subtle problems with code that’s developed on a Windows machine, for example, when it is run on a Unix machine. For example, the two systems use different newline characters.
- We will use the turnin program on lectura to hand in assignments.

Prerequisites, Required Knowledge

- **Prerequisites**: CSC 127B or CSC 227.
- You need to be a competent programmer in a procedural/object-oriented language, such as Java or C.

Now What?

Let’s Have Fun!!^a

^aThat’s right—learning new languages is fun!