CSc 372

Comparative Programming Languages

0 : Administrivia

Christian Collberg

collberg+372@gmail.com

Department of Computer Science

University of Arizona

Copyright © 2005 Christian Collberg

Contact Information

Class	:	372 — Comparative Programming Languages
Lecturer	:	Christian Collberg
Email	:	collberg+372@gmail.com
IM	:	csc372 (AIM)
WWW	:	http://www.cs.arizona.edu/ collberg/Teaching/372/2005
Office	:	Gould-Simpson 758
Office Hours	:	Open door policy
Phone	:	621-6612
Lectures	:	MWF 2:00-2:50, GLD-S 906
Honor's section	:	M 4:00-4:50, GLD-S 805
TA	:	TBA

Course Outline

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

Syllabus

You are responsible for reading

and understanding this syllabus.

If you have any concerns or issues

about the information in this document

you should bring them up during the

first week of class.

Course Description

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 the languages: 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.

Exam-Schedule

gives the final exam schedule.

1. The midterm exam is scheduled for Fri, Oct 14. This may change, so pay attention in class and check the web site.
2. The final exam is scheduled for Wed, Dec 14, 2:00-4:00. Check

   http://www.registrar.arizona.edu/schedule054/exams/054exams.htm

   for any changes.

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 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.

Course Methodology

- 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.

Course Methodology...

- 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 on your own.

Required extracurricular activities

- Programming assignments.
- Working programming exercises on your own.

Special materials required for the class

- None.

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.

Assessment Scheme

Tests, Quizzes, and Assignments

There will be

1. one mid-term test worth a total of 10%;
2. one comprehensive final exam, worth a total of 20%;
3. five unannounced quizzes, where the lowest score will be dropped, worth a total of 10%;
4. two-to-four assignments on Haskell, worth a total of 20%;
5. two-to-four assignments on Prolog, worth a total of 20%;
6. two-to-four assignments on Icon, worth a total of 20%.

Late Assignments

- Assignments handed in no more than 24 hours late will incur a 10% penalty.
- Assignments handed in more than 24 but no more than 48 hours late will incur a 20% penalty.
- Assignments handed more than 48 hours after the deadline will receive a grade of 0.

Making up Tests

You cannot make up the midterm or final exam unless

1. you have notified the instructor in writing (email is fine) or by phone prior to the test that you will be absent, and
2. you receive permission from the instructor to take the test at a later date.

Pop quizzes will be given out at the beginning or end of class and cannot, under any circumstance, be made up at a later date.

Curving

- All grades (for exams, quizzes, and assignments) will be curved up by throwing away the highest grade in the class and scaling up such that the second highest grade is 100.
- The curving is done to adjust for particularly difficult tests/assignments, and to prevent an outlier from skewing the grade distribution.
- You cannot, after scaling, receive more than 100 on any exam, quiz, or assignment.

Grade Assignment

- You will fail the class if you get less than 50 (after curving) on the final exam.
- Otherwise, a curved total grade of [90,100] gives you an A, [80,89] a B, [70,79] a C, [60,69] a D, and 59 and below an E.

Incomplete work policy

- Except under exceptional circumstances I will not assign incomplete grades.
- I decide what is an exceptional circumstance.

Detailed Grading Scheme

- To avoid any ambiguities, I have formalized the informal rules given above.
- The rules below should be considered *minimum* requirements to achieve a particular grade. The instructor reserves the right to do additional adjustments, as necessary.
- Any contradictions, omissions, errors, or ambiguities in the grading scheme will be resolved by the instructor.
- Any issues or concerns regarding the grading scheme should be brought to the attention of the instructor within the first week of class.

Details — Curving

- All raw scores range from 0 to 100.
- Each individual score (final, midterm, quizzes, assignments) will be curved using the function

$$\mathrm{curve}(x, s) = \min(100, (100.0/\max(x - \max(x)))x_s)$$

  where x is a set of scores (for an assignment, a test, etc.) and s is a student.
- Note: $-$ is set subtraction.
- $\mathrm{curve}(x, s)$ returns s's score, curved up by $100.0/2nd_highest_class_score$.

Details — Curving...

- For example, assume the following final exam scores:

$$34 \quad 45 \quad 66 \quad 88 \quad 98$$

After the curve has been applied, the scores will be

$$38.6 \quad 51.1 \quad 75 \quad 100 \quad 100$$

Details — Exams

final exam:

- Let f be the set of final exam scores.
- Let f^s be the final exam score for student s.
- Let $\mathcal{W}^f$ be the weight of the final exam (20%).
- $t_f^s = \text{curve}(f, s)\mathcal{W}^f$ is the curved final score for s.

midterm exam:

- Let m be the set of midterm exam scores.
- Let m^s be the midterm exam score for student s.
- Let $\mathcal{W}^m$ be the weight of the midterm exam (10%).
- $t_f^s = \text{curve}(m, s)\mathcal{W}^m$ is the curved midterm score for s.

Details — Quizzes

- Let q_i be the set of scores for the i:th quiz.
- Let ${q_i}^s$ be the score for student s on the i:th quiz.
- Let $\mathcal{W}_i^q$ be the weight of the i:th quiz ($\mathcal{W}_i^q = 10\%/4$).
- We throw away each student's lowest quiz score (note: $-$ is set subtraction):

$$q'^s = q^s - \min(q^s)$$

- $t_q^s = \sum_i (\text{curve}(q'_i, s)\mathcal{W}_i^q)$ is the total curved quiz score for student s.

Details — Assignments

- Let a_i be the set of scores for the i:th assignment.
- Let a_i^s be the score for student s on the i:th assignment.
- Let $\mathcal{W}_i^a$ be the weight of the i:th assignment ($\sum_i \mathcal{W}_i^a = 60\%$).
- Let α_i^s be the assignment score after late penalties have been applied:

$$\alpha_i^s = \begin{cases} a_i^s & \text{if the assignment is handed in on time} \\ 0.9a_i^s & \text{if the assignment is } > 0 \text{ and } \leq 24 \text{ hours late} \\ 0.8a_i^s & \text{if the assignment is } > 24 \text{ and } \leq 48 \text{ hours late} \\ 0 & \text{if the assignment is } > 48 \text{ hours late} \end{cases}$$

Details — Assignments...

- $t_a^s = \sum_i (\text{curve}(\alpha_i, s)\mathcal{W}_i^a)$ is the total curved assignment score for student s.
- If, for whatever reason, the actual number of assignments is less than the planned number, the $\mathcal{W}_i^a$'s will be scaled up uniformly.

Details — Total Scores

- The raw total score for student s is

$$t_s = t_f^s + t_m^s + t_q^s + t_a^s$$

- We do a final curving of the total and round up to the nearest integer:

$$\text{total}_s = \lceil \text{curve}(t, s) \rceil$$

Details — Grade Assignment

- The final grade assignment for student s is

$$\text{grade}_s = \begin{cases} E & \text{if } t_f^s < 50 \\ \begin{cases} A & \text{if } \text{total}_s \in [90, 100] \\ B & \text{if } \text{total}_s \in [80, 89] \\ C & \text{if } \text{total}_s \in [70, 79] \\ D & \text{if } \text{total}_s \in [60, 69] \\ E & \text{if } \text{total}_s < 60 \end{cases} & \text{otherwise} \end{cases}$$

- In other words, a student with a curved final exam score $t_f^s < 50$ will fail the class, regardless of their results on the other assessment categories.

Policies

Office hours

- I use an open door policy:

Collberg's Café

- Please come and see me to chat, ask questions, or snack:

Attendance Policy

- My goal is to keep class attendance high so that we can get good discussions going in the class.
- 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.
- To encourage class attendance and participation there will be five unannounced pop quizzes during the semester.

Attendance Policy...

- The honors section has mandatory attendance. We will probably not meet every week, but when we do meet, you have to be there.
- All holidays or special events observed by organized religions will be honored for those students who show affiliation with that particular religion. Absences pre-approved by the UA Dean of Students (or Dean's designee) will be honored.

Subject to Change Policy

- The information contained in this course syllabus, other than the grade and absence policies, may be subject to change with reasonable advance notice, as deemed appropriate by the instructor.
- The instructor reserves the right to
  1. add, drop, or change topics;
  2. 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

- There is no objectionable material in this class.

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://www.salt.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…

- 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.

Student Code of Academic Integrity...

- 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://studpubs.web.arizona.edu/policies/cacaint.htm` the University of Arizona Code of Academic Integrity.

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

Expected classroom behavior

- Be courteous and treat others in the class with respect.
- Please be courteous to other students by refraining from talking, playing loud music in your headphones, silencing cell phones, pagers, etc.
- We come to class to learn: don't read the newspaper, solve cross-word puzzles, etc.

Policies against threatening behavior

- Read and abide by the following link:

  http://policy.web.arizona.edu/~policy/threaten.shtml.

Now What?

Let's Have Fun!!![a]

[a] That's right — learning new languages is fun!