Principles of Programming Languages

Dr. Mark C. Lewis
Trinity University
Fall 2004

Opening Discussion

- Have you read the first chapter?
- Why are you taking this course? What do you think we are going to learn about? What do you want to learn about? How would you characterize this course?

Course Web Page

- The web page for this course is at www.cs.trinity.edu/~mlewis/CSCI3368-F04
- This has the syllabus and a link to a schedule. The lecture notes will be linked to from the schedule before class every day.
- We meet here every TR unless otherwise marked on the schedule.
- Text: “Concepts of Programming Languages” by Sebesta. You'll read the whole thing this semester.

Course Description

- This course probably isn't very well understood in the department. The way it is taught has varied over the years and from college to college.
- This is not a class where you learn a whole bunch of different languages. A shallow knowledge of 20 languages doesn't benefit you much.
- This class is about the fundamental concepts that go into programming languages. Understanding those allows you to more quickly learn new languages and makes you better at the ones you already know.

Basic Course Outline

- The schedule gives you an idea of what we will be covering over the course of the semester. Basically we are going straight through the book.
- You will be evaluated on four areas.
  - Assignments : 25%
  - Tests : 20%
  - Class Participation : 25%
  - Final Project : 30%

Assignments

- Each chapter has a set of problems and programming problems at the end. You will need to pick 4 of these to do and hand in to me the class period after we finish each of the first 15 chapters.
- Originality in selecting problems is rewarded (+10 points if you are the only person to do a problem) and lack thereof is penalized (-10 points if more than 1/3rd of the class does that problem).
- You can work with others, but must turn in your own work.

Tests

- There will be two tests, a midterm and a final. Each counts for 10% of your grade. They are both intended to be roughly an hour long.
- These aren't meant to kill you and notice how little they count towards your grade. However, they are my way of checking that you understand the concepts independent of your other classmates.

Class Participation

- It might seem odd having 25% of your grade be class participation, but for this class I want you to be engaged in significant discussions instead of just listening to me lecture.
- You must do the readings before you arrive and show up with two questions/comments about material in the reading. The quality of those comments will go into calculating your class participation grade for each class along with what you actually say in class.

Final Project

- The single largest part of your grade will come from a project that is due during the finals period. The schedule lists important "early" due dates for the project.
- For this project you can choose to do almost anything significant related to programming languages.
  - Design your own language and argue for choices made in the design.
  - Detailed analysis of existing languages.
  - Active research in PL.

Programming Domains

- Coding can be broken into different domains.
  - Scientific Applications
  - Business Applications
  - Artificial Intelligence
  - Systems Programming
  - Scripting Languages
  - Special-Purpose Languages
- What languages do you know? What domain are they in? What problems do you use them for?

How We Evaluate Languages

- Readability
  - Simplicity/orthagonality, Control structures, Data types & structures, Syntax design
- Writability
  - Above plus: Support for abstraction, Expressivity
- Reliability
  - Above plus: Type checking, Exception handling, Restricted aliasing
- Cost

Influences

- Machine architecture
  - This has had a HUGE influence on the design of programming languages? Why? How?
- A language is also impacted by the type of methodologies it allows. Alternately we might say by the paradigms that it supports.
  - Imperative
  - Functional
  - Logic
  - Object-oriented

Trade-offs

- The evaluation criteria we use often are at odds.
- What are some examples of how they are at odds? How do these things manifest in the languages that you know? Which types of trade-offs do you think take priority? What other variables matter in the trade-offs?

Implementation Methods

- There are three main ways of implementing a language
  - Compiled
  - Pure Interpreted
  - Hybrid
- Name some languages that do each. What are the trade-offs of each?

Programming Environments

- Some of the biggest advances in programming haven't actually dealt directly with the languages. Instead, they have dealt with the way that we write those languages.
- What is your favorite programming environment? Why do you like it? What are the strengths of it?

Numerical Code Demonstration

- I have a small code that does a numerical integration of bodies under the influence of gravity. It puts a central “star” down with 100 smaller bodies around it. I wrote in it Java and ported it to C++ to do some speed tests.
- Let's look at how the two compare.