

CS 655
Graduate Core Course on
PROGRAMMING LANGUAGES

Prof. Paul F. Reynolds, Jr.

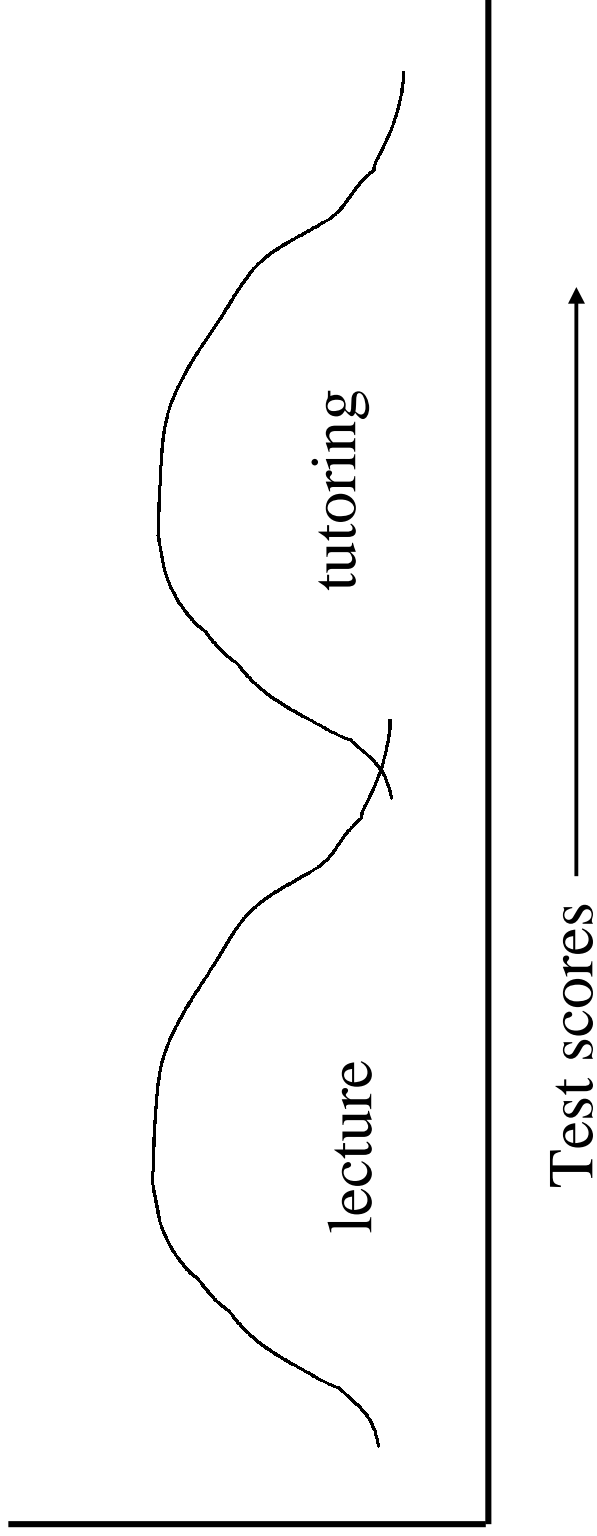
Department of Computer Science
Olsson 214
(804) 924-1039

reynolds@virginia.edu

655 Roster

- Sign up today!
- Give:
 - name
 - school/dept/year
 - languages known
 - e-mail address (skip the ‘virginia.edu’ part)

Delivering the Goods



Course Details

- (recommended) text:
 - *Programming Lanuages: Concepts & Constructs*
 - Sethi
 - Benjamin/Cummings, 3rd Edition, 1996
- Grading:
 - Homework - 15%
 - Midterm/final - 20% / 30%
 - Project - 25%
 - Class Participation - 10% (taken seriously)

Support

- Office Hours/Out of Class Communication:
 - Tues/ Thurs 3:30 - 5:00, *and by appt.*
 - E-mail read frequently (reynolds@virginia.edu)
 - CS655 homepage exists; check it daily;
 - <http://cs.virginia.edu/CS655>
- TA:
 - Jim Gunderson
 - Gunderson@virginia.edu

Exams

- Midterm (take-home, open book, pledged):
 - handed out in class on Tuesday, 23 March
 - collected at my office by 5PM, Thursday, 25 March
- Final (take-home, open book, pledged)
 - Handed out last day of class (Tues, 04 May)
 - collected at my office, data TBA.

(more on) Exams

- Take-home
- Open book
- Pledged
- Essay
- Emphasis on problem solving
- Best preparation is weekly writing assignments, and anticipating questions
- (I'd rather be giving multiple choice...)

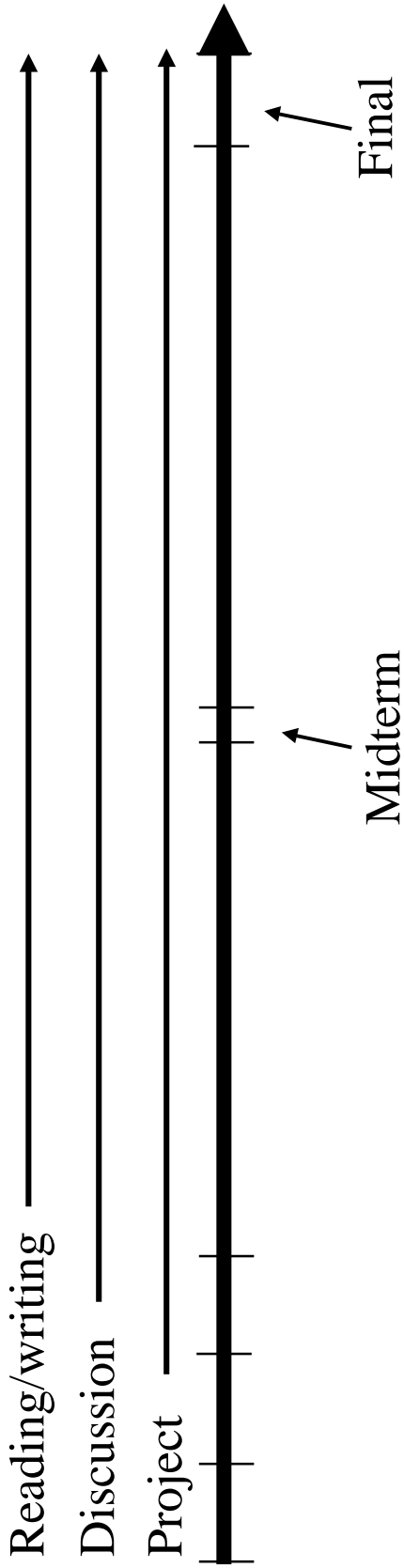
Pledged Work

- Review University honor code
 - We will abide by it.
- OK to talk about readings, translators, etc.
- Not OK to show another your work.
- Not OK to discuss answers.
- Be sure to cite sources!
 - Plagiarism is a serious offense

Project

- Analysis of selected set of languages.
- Requires installing translators.
- Requires reading language documents.
- Can be done in groups (encouraged).
- Opportunity to do programming.
- Assignment will be given in next few days.
 - Tell me what you'd like it to be...

Semester Timeline



Sources

- Libraries (CS, Sci Tech, Alderman)
- Indices (Virgo, ACM CDRoms, Inspec, webSPIRS)
- Web: browsers and search engines
- ACM / IEEE Journals:
 - CACM Transactions on Computers
 - JACM Transactions on S.E.
 - TOPLAS Trans on Par & Dist Computing
 - SIGPLAN Computer
 - Surveys Software
 - Letters on Prog Langs

Assignment For Tuesday, 26 Jan 99

- Read (papers in file drawer by copy machine):
 - Dijkstra: “Threats...”
 - Hoare: “Hints”
 - Wegner: “...First Years...”
 - Ousterhout: “Scripting, Higher Level Prog...”
 - Backus: “Functional Programming”
- Write:
 - One page paper on what you expect to get out of cs655 and why
- Explore:
 - Prove you’ve visited CS, SciTech, Alderman libs

How Do I Write a Reaction Paper?

- Read the assigned paper(s).
- State a thesis and attempt to substantiate it.
- Limit summary of assigned paper(s) to a paragraph or so (each).
- Make a few points and defend them well.
 - (as opposed to lots of topics with no depth)
- Draw on unique aspects of your experience.
- Don't be afraid to be bold.

Course Goals

- Expose to:
 - History
 - Design principles (let's not repeat mistakes)
 - Past and current issues in PL design
 - Introduction to major current research areas

Philosophy

- Roots and wings
- “Those who cannot remember the past are condemned to repeat it.”
 - Santayana: *Life of Reason*
- “Language is thought”
 - Whorf-Sapir Hypothesis

Why Study Programming Languages?

- Better ability to critique languages:
 - Need in order to select language wisely.
 - Need in order to identify what's truly novel and useful.
- Better Ability To Design Languages:
 - Those who ignore history are bound to repeat it...
- Advancement Of Computing:
 - One Day We Might Get It Right, Until Then...

What Is It We're Studying?

- High level languages
 - vehicles for expressing algorithms
 - vehicles for directing a computer to solve problems
 - broader view: includes O.S. Command Languages, DBQL's, GUI's, Spreadsheets, Visual Languages, Scripting Languages
- Tend to focus on the former type in this class
 - must remain aware of the influence of the latter!
- What's a 'very high level' language?
- What distinguishes HLL's from Low LL's?

What Is It We're Studying (Rebelsky)?

- What is a programming language?
 - Hoare: (In part): A tool to aid the programmer.
 - Louden: A notational system for describing computation in machine-readable and human-readable form.
 - Rebelsky (Dartmouth): A notation for formally expressing algorithms so that they may be understood by humans and computers.
 - Reade: One, rather narrow, view is that a program is a sequence of instructions for a machine. We hope to show that there is much to be gained from taking the much broader view that programs are descriptions of values, properties, methods, problems, and solutions. The role of the machine is to speed up the manipulation of these descriptions to provide solutions to particular problems. A programming language is a convention for writing descriptions which can be evaluated.
 - Stansifer: The purpose of language is communication. [...] Programming languages are used to communicate with literal-minded machines.

History

- Where Did All These Languages Come From?
 - The Difficulty Of Programming Computers
- Are They All Needed? Will There Be More?
- Pseudocodes (195X) - Many
- FORTRAN (195X) - IBM, Backus
 - Led To FORTRAN I, II, III, IV, 66, 77, 90
- LISP (196X) - McCarthy

History (2)

- ALGOL (1958) - Committee
 - Led To ALGOL 60, ALGOL W, ALGOL 68, Pascal, Ada
- COBOL (196X) - Hopper
- PL/I - IBM
- Functional Programming - FP, Scheme, Haskell, ML, OCML
- Logic Programming - Prolog, GHC
- Object-oriented - Smalltalk, C++, Python, Java, Sather, Eiffel
- Parallel / Non-deterministic Programming
- Aspect-oriented Programming
- Visual Programming

What Is The Future Of HLL's?

- Will we always have a plethora of languages?
 - 1000's! (> 2300 in 1996)
- Yes
 - History shows proliferation is only getting worse.
- No
 - Standards and paradigms will prevail
- ???

What Influences Language Design?

- Architectures
- Algorithms
- Others
 - security
 - verifiability
 - large-scale programming
 - programmer productivity
 - specific applications
 - generality and standardization
 - implementation issues

Lampert: design of languages is not as important as design of algorithms and hardware. As latter are developed, languages will adapt to support them.

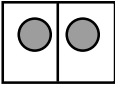
- Agree???

How Do You Think?

- Shortest path-
 - d_{xy} : distance from node x to node y
 - $sd_A(y)$: shortest distance from node A to node y
- Problem: express solution to finding distance from node A to all other nodes, y , in graph

$$sd_A(y) = \text{Min} (sd_A(y), sd_A(x) + d(x,y)) \text{ all } x,y$$

Why There Will Never Be Just One Language

- APL:  monadic matrix inversion operator
- ML, Haskell: type inferencing
- Icon generators
- Unity non-determinism
- Perl emphasis on strings
- Java emphasis on interpretation

Weaknesses are often found in conflicting features

Truisms?

- Application-specific languages will always exist
- Large, design-by-committee languages tend to fail
- Small, lone ranger languages tend to succeed.
- Complexity of a language tends to affect its
 - readability
 - writability
 - translatability
 - portability
 - predictability

Next Time(s)

- Paradigms
- Principles