CS655

TTh 12:30-13:45 **MEC216** **Spring, 1999**

COURSE TITLE: Programming Languages

REQUIRED BACKGROUND: Undergraduate course on analysis of programming languages. Working knowledge of the equivalent of an undergraduate text such as Sebesta's: Concepts of Programming Languages Benjamin/Cummings, 2nd Edition, 1993 or 3rd Edition, 1996.

TEXTBOOK: (Recommended, not required) Sethi, R. *Programming Languages, Concepts and Constructs,* 2nd Edition, Addison-Wesley, 1996.

COURSE DESCRIPTION: The goal of this course is to ensure you are well informed about 1) the history of programming languages, 2) design issues affecting languages and 3) current research issues in the design and implementation of programming languages. The history of high-level language design will be a thread running through the course. We will look at design principles, establishing a working set of principles that will then apply to better known high-level languages. Also, we will focus on current research issues, including formal semantics, verification, object-oriented languages, meta-classes, polymorphism, exception handling, very high-level languages, non-determinism and parallel languages.

This will be a papers-oriented course.

INSTRUCTOR: Paul F. Reynolds, Jr.

OFFICE: Olsson 214
HOURS: TTh 3:30-5:00PM, & by appointment (reynolds@virginia.edu, 924-1039). E-mail entertained anytime. Also, there is a homepage for CS655; check it daily: **http://www.cs.virginia.edu/CS655**

TA: Jim Gunderson (gunders@virginia.edu). Hours by appointment, E-mail entertained anytime.

ASSIGNMENTS: Numerous reading assignments, homework assignments and a project. In general you will be asked to write short response papers to the papers you read. Also, problems related to papers will be assigned. The project will be analysis oriented; you will be writing and analyzing programs you write in some of the languages we investigate.

EXAMS: A midterm and a final. Midterm will be T/TH, 23-25 March (take-home).

EVALUATION: The midterm counts 20%, the final counts 30%, homework is 15%, the project is 25% and class participation is 10%. Not all homework assignments will be graded, but you are expected to do all assigned work. Also, you are expected to be prepared for class discussions; doing the homework helps.

PLEDGE POLICY: All work in this course will be pledged. If you are not familiar with the university's pledge policy, contact a student honor advisor or see me. Essentially, all work should be your own. You are to work alone unless I give you explicit permission to work in collaboration with others. The project will be a pledged, collaborative effort. In all work, please cite appropriate references (e.g., any ideas that are not completely your own).

Also, unless told otherwise, you may make full use of library resources, including the web. However, do not remove material from libraries if it would be a potential detriment to others in the class (for example, if the library only has one remaining copy). Either use the resource there, or copy it.

TOPICS:

- Historical overview
- Language paradigms
- Design principles
- Language analysis:
  - FORTRAN(s)
  - ALGOL60
  - ALGOL68
  - PL/I
  - Pascal
  - C
  - Icon
  - CLU
  - LISP
  - Scheme
  - Ada83/Ada95
  - C++, Smalltalk, Python, Sather, Eiffel
  - Prolog
  - ML, Haskell
  - Java, Self
  - Unity
- Interesting research areas:
  - Language type systems, polymorphism, type inference
  - Meta-classes and other advanced object oriented language features
  - Pattern matching and lazy evaluation
  - Garbage collection and memory management
  - Reliability features, such as exception handling and "design by contract"
  - Formal semantics and mathematical models
  - Program verification

FOCUS: This course can be fun. I do my best to make it that way by exposing you to those aspects of language design that are fun to think about. If there's a topic you'd like to see covered, or if there's an approach to exploring issues you'd like to take, please feel free to speak up. In general, the more you get involved in the class, the better it is for all of us.

There is a heavy emphasis on reading, reacting and writing in this course. My thought is that these skills are good ones to sharpen, and this course is a good place to do it. As a result, programming is de-emphasized somewhat (we can't do it all). Some students would rather have more programming and less reading and writing – I know this from years of reading through course evaluations – the project is meant to be your opportunity to experiment with using languages. If programming is your forte, use the main project as your opportunity to exploit it. And, of course, you are encouraged to explore any language we may be covering on your own.

The web has made access to translators, as well as discussions about just about every language that's ever existed, quite easy. I encourage you to use the web resources to their fullest extent, for all of the work you do in this class. Be sure to treat the web as you would (should) any other written source of material: if you use it, cite it!