

Proposed Course Outline

**AMOD 501H: Advanced Programming Language Concepts
(based on COSC 410H: Comparative Programming Languages)**

Course Description:

This course develops a conceptual framework to critique and compare the different features and types of high-level programming languages. It is not a survey course on programming languages, but rather a course that focuses on the key choices that go into the design of a programming language and how they are manifested in so many different (and not necessarily sound) ways. It is therefore a course that gives the student the wherewithal to ask important questions such as:

- Is this language a good choice for its intended purpose?
- What are the trade-offs of this language with respect to efficiency, readability, writeability, and security?
- What are the features of this language and are they intuitive, reliable, and maintainable?
- Is the language equipped with the tools I need?

The course will spend equal time on the fundamental concepts of procedural and object-oriented languages as it does on the more advanced topics of concurrency, exception handling, functional languages, and logic languages.

Main Text:

Robert W. Sebesta, *Concepts of Programming Languages* (7th Edition), Addison-Wesley, 2006
All chapters will be covered (except Chapters 2, 3 and 4)
Additional readings for the discussion paper and term project

Grading Scheme:

Test 1 (covering Chapters 1, 5-12)	30%	
Test 2 (covering Chapters 13-16)	30%	
Discussion Paper (as described below)	10%	
Term Project (as described below)		
Written report	20%	
Oral presentation	10%	
	30%	
Total		100%

Discussion Paper:

The discussion paper is a 10-12 page report that examines the following topic:

“Parallelism and functional programming languages”.

Drawing on both theoretical arguments and experimental languages such as parallel Haskell and NESL, the paper is expected to outline the key design and implementation issues that make functional languages particularly well- or ill-suited for parallel computation.

Term Project:

The general objective of the term project is to design and present your own high-level (non-parallel) programming language. But before one can sit down and design a programming language, it is absolutely necessary to answer three critical questions:

- 1) Who is the intended audience (users)?
- 2) What are the intended applications?
- 3) What type of language will satisfy this audience and its applications (logic, functional, object-oriented, graphical, imperative, scripting, and so on)?

The answers to these three questions will have a significant impact on your design decisions and will influence the feasibility and choice of data types, control statements, subprograms, exception handling, concurrency, and so on.

Your final report will therefore be a concise, clear, and comprehensive description of your new programming language, complete with syntax and semantic definitions (though they may be informally stated). Use examples and diagrams whenever possible to clarify your language features. Include as well, snippets of code to show how the language "looks".

In essence, your report is the basis of a new language manual. As a student of computer science and as one who has learned programming languages using texts and manuals, reflect on how you would present a new language to the intended audience. Reflect as well on how you would address those features of a language which you have found to be particularly annoying or non-intuitive.

This project will be deceptively challenging. Do budget sufficient time to consider the design issues and to create a programming language that is consistent (orthogonal), well-thought out, and if possible, unique. Use your imagination and try to think "outside the box".

The report which is expected to be 20-25 pages makes up 2/3 of your grade on the term project. The remaining 1/3 is based on a 30 minute oral presentation to the class. The presentation will be evaluated based on preparation and professionalism as well as the generated interest in your project. The presentation is expected to focus principally on those features of the language which define its special appeal.

Additional Work:

This course is offered independently as a reading course and without undergraduate participation. It does cover a fair bit of the material of COSC 410H, but with a greater emphasis on the latter chapters. These chapters delve into the more advanced concepts such as concurrency, exception handling, functional programming, and logic programming. The paper on parallelism and functional programming languages is an extension of this material. It is also expected that the term project on the specification of a new programming language will present a reasoned and fairly comprehensive look at a "programming language deficiency" and how it can be addressed. This work also will be presented orally to fellow graduate students and faculty.