CSCI 5232 PROGRAMMING LANGUAGES and Compiler Design

Description:
Study of various programming languages from conceptual standpoint; topics will include formal language definition, data storage techniques, design techniques and implementation issues for compilers. Both numeric and string processing languages will be covered. The course assumes knowledge of at least one imperative language.

Pre-requisites: Consent of the instructor

Audience:
This course is designed for students in the graduate program in computer science.

Instructor: Dr. Saïd Bettayeb
Ph. 283-3857
Rm. Delta 166

Office Hours:  Mon. 5:00 - 7:00 p.m. and Wed. 2:00 - 4:00 p.m.
Or by Appointment

Teaching Assistant: Mr Rahul Chalavadi
TA’s Office Hours:  Mond. 4:00 - 7:00 pm
                   Tues. 4:00 - 10:00 pm
                   Thurs. 9:00 - 2:00 pm

Text: Programming Language Concepts,
Edition: 3rd
Authors: Carlo Ghezzi & Mehdi Jazayeri
Publisher: John Wiley
ISBN: 0-471-10426-4

Prerequisites: Programming in Pascal, C, PL I, C++, Java, Fortran or Ada

Grading Policy:
Midterm Exam 35%  October 12, 2009
Final Exam 45%  Dec. 7, 2009
Homework, Programming Assignments and Quizzes 20%
Specification of Course and Objectives

Statement of General Goals and Objectives:

The student learns in this course the principles of programming languages. The student will learn the skills required to design and implement programming languages. The various paradigms in programming language designs are covered as well as techniques for Compiler Design.

Upon completion of this course, students will be able to:

- Understand the overall structure of programming languages, principles of language design, and programming language evaluation criteria.
- Contrast between imperative languages and functional languages
- Understand the concept of exception, how to raise, its scope and handler.
- Use and Understand programming in Logic Paradigm (prolog)
- Understand design and implementation issues of programming languages
- Write small programs in LiSP and Prolog
- Understand the phases composing a compiler.
- Understand how the intermediate code is generated and the process of optimizing it.
• Understand what a parser is and the various kinds of parsing techniques.

COURSE OUTLINE with a Tentative Schedule

I. Programming Languages     August 24-29, 2009
  1. Definition and use
  2. Evolution
  3. Issues and perspective of the course

II. Functional Programming:  August 31 – Sept. 12, 2009
  1. Features of functional languages
  2. LISP, ML

  1. PROLOG

IV      Compilers and Interpreters   Sept. 21-26, 2009
  A. Phases of the compiler
  B. Lexical Analysis:
     i. Tokens and Lexemes
     ii. Buffering
     iii. Finite State Automata
     iv. Regular Expressions

  a. Grammars
  b. Top-Down Parsers
  c. Bottom-Up Parsers
  d. The Operator-Precedence Parsers
  e. The LR-Parsers

   October 12, 2009 Midterm Exam.

VI. Main Features of Programming Languages   Oct. 19 – Nov. 14, 2009
  1. Overall structure
  1. Variables
     a. declarations - scope - lifetime - value - type
     b. concepts of bindings
  2. Data Types
     a. builtin types
     b. user-defined types
     c. issues associated with types -- typing mechanisms
     d. type inference
  3. Control Structures
     a. statement-level control structures
        * sequencing
        * iteration
        * selection
     b. unit-level control structures
        * procedures
          + parameter passing mechanisms
          + block-structure
        * exception handling mechanisms
4. Abstract Data Types
   a. concept
   b. theoretical basis (heterogeneous algebras)
   c. algebraic specification
   d. language constructs

VII. Storage Management  Nov. 16-24, 2009
    1. Storage allocation
       a. variables
       b. stack-based allocation
       c. activation records
       d. display vectors
       e. heap

VIII. Object-oriented languages: C++: Nov. 30-Dec. 5, 2009

Final Exam Dec. 7, 2009

**Accommodations** (as specified by the Americans with Disabilities Act) - If you will require special academic accommodations, please contact the [Disability Services Office](http://prtl.uhcl.edu/portal/page/portal/PRV/FORMS_POLICY_PROCEDURES/STUDENT_POLICIES/Academic_Honesty_Policy) at 281-283-2627.

Academic Honesty Code: see section 2.1.4 in the Students Life Policies handbook for the UHCL Academic Honesty Code.