CPSC 320

Programming Languages

UNBC
Fall 2007

Who, Where, When?

Rooms	Lectures are in 5-159
Hours	14:30–15:50 T R
Web page	`http://web.unbc.ca/~casper`
e-mail	`casper@unbc.ca`
Instructor	David Casperson
Office	T&LC 10-2046
Telephone	960-6672

Grading:

Assignments	:	20%
Quizzes	:	up to 5%
Exam 1	:	20%
Exam 2	:	20%
(Final) Exam 3	:	35%

Note:
I don't yet know what exactly what the format will be for the assignments for this course, so the weighting of the assignments might change after class discussion.

Prerequisites:
C^- or better in CPSC 281, (and by transitivity CPSC 100, CPSC 101, CPSC 141, CPSC 142, and CPSC 200).

Text:
Programming Languages: Principles and Practices. by Kenneth C. Louden. **Second Edition.**

Dates:

Exam 1	:	2007-10-09 T
Thanksgiving	:	2007-10-11 M, *U closed*
add/drop	:	2007-10-16 T
off campus	:	2007-10-25 R
	:	2007-11-12 M, *U closed*
last class	:	2007-11-29 R
Exam 2	:	2007-11-01 R
(Final) Exam 3	:	*in* 12-05 to 12-17.

What? Topics chosen from (*not necessarily in the order listed*):

- Design principles. Simplicity, abstraction, orthogonality, reliability, ...
- Syntax: Lexical conventions and analysis — tokens, concrete syntax, grammar descriptions, derivation trees, abstract syntax.
- Basic semantics: bindings, scope, environments, allocation, lifetime. Constants, variables, and pointers. Aliases, dangling references, and garbage.
- Formal semantics: Operational, Axiomatic, & Denotational Semantics.
- Data Types. Simple Types. Mathematical models. Type constructors and standard non-simple types. Type equivalence and type checking. Polymorphism.
- Control Structures. Selection, looping, and non-local flow. Procedures and Environments. Recursion. Parameter-passing mechanisms. Exception handling. Continuations.
- Programming in the large. Modules and packages. Information hiding, data abstraction. Object-based and object-oriented programming.
- Programming Language Paradigms. Object-Oriented, Functional and Logic Programming. Mathematically-modelled languages. Exotic languages.

Why?

Language popularity and availability changes constantly over time, forcing most programmers to learn multiple languages and paradigms. This course introduces introduces general ideas that underly programming languages and their design and description, giving a framework for reasoning about, learning, and designing computer languages.