Programming Languages

Web-page: `http://casper.unbc.ca/Semesters/2018/320.php`

Prerequisites: A grade of C^- or better in CPSC 242, and CPSC 200; or permission of instructor.

Accommodations: If there is any student in this course who, because of a disability, may have a need for special academic accommodations, please come and discuss this with me, or contact the Access Resource Centre located in Teaching & Learning 10-1048.

Who, Where, When?

Rooms	Lectures are in 5-183 Labs are in 8-361
Hours	14:30–15:50 T R
Web page	`http://casper.unbc.ca/`
e-mail	`David.Casperson@unbc.ca`
Instructor	David Casperson
Office	T&LC 10-2080
Telephone	(250)960-6672

Text: Recommended *Programming Languages*. by R. W. Sebesta. (10/11)th Edition. There is no required text.

Grading and Dates:

Participaton :		: 10%
Homework :		: 20%
Thanksgiving :	Mon, Oct 08	
Exam 1 :	Thu, Oct 18	: 25%
last drop :	Thu, Oct 25	
Remembrance hol :	Mon, Nov 12	
Exam 2 :	Tue, Nov 20	: 25%
Language Presentation :	???	: 20%
Course eval. :	Thu, Nov 22	
last class :	Thu, Nov 29	

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.

Updated September 4, 2018