CMPUT325: Non-Procedural Languages

Instructor: R. Greiner GSB 621 x5461
Office Hours: (immediately after class, or by arrangement)
http://www.cs.ualberta.ca/~greiner/C-325

TAs:

Julian Fogel	julian	@cs.ualberta.ca
Tim van Allen	vanallen	@cs.ualberta.ca
Dave Gomboc	dave	@cs.ualberta.ca
Marc Perron	perron	@cs.ualberta.ca

Textbooks: "*Functional Programming in Lisp + Prolog Intro*"
H. Levesque and R. Reiter, 1987.

Lecture Notes: http://www.cs.ualberta.ca/~greiner/C-325/Notes/index.html

Rec'ded: "*Prolog Programming for Artificial Intelligence*", (2
I. Bratko, Addison-Wesley, 1990.

"Szafron Notes"

"*The Architecture of Symbolic Computers*"
P. Kogge, McGraw-Hill, 1991.

Grading:

4 assignments	(40 points total)
MidTerm	(25)
Final	(35)

Computer, Homework

- Accounts on { obed, gsb001, ... }
- Post questions to `c325`
  Read answers (+ news) using
  `news:ualberta.courses.cmput.325`
- Assignments due via `try`
  NO late assignments accepted!
- Work should be
  - Correct
  - Well documented
  - Convincing (include examples, tests)
- `try` is MIS-NAMED!
  think "`submit`"

Warnings

- Do not use anyone else's program.
  To avoid imputations of cheating:
  Discuss only *general approaches* to problem.
  Do not take written notes.
- Do not
  interfere with the
  operation of the computing system
  interfere with other's files
  change another's password
  copy another's program
- SERIOUS:
  Punishment ranges from 0 in course to expulsion.

Lab Times

	M	12:30pm	5	Dave
x	Tu	8am	8	Julian
	Tu	5pm	10	Julian
x	W	8am	2	Dave
	W	2pm	9	Dave
	W	5pm	9	Dave
	Th	2pm	9	Julian
	Th	5pm	10	Julian
x	F	8am	2	Tim
x	F	5pm	3	Tim

Only CERTAIN WEEKS:
+ for IntroLisp: 13-17/Sept
+ for IntroProlog: ??[tentatively 18-22/Sept]
+ just 1weeks before HWs due

Only FIRST HOUR
Otherwise, by arrangement!

Tests/Final

- Mid-term will be
  Thurs 28/Oct, at 7pm
  in Room V107!

- Final exam cannot be rewritten.

- If you apply for a deferred final exam: the only time for the deferred final exam is 7 Jan 2000 at 2:00pm.

Standard Programming Languages

Examples: FORTRAN, Turing, ADA, PASCAL, C, PL/I, ALGOL, COBOL, assembly, ...

Imperative: Based on COMMANDS

- *Assign* value to variable, array element, ...
- *Go to* location
- *Return* from procedure
- *Read/Write*

+ Composite Commands

–	Sequence	`begin ... end`
–	Conditional	`if ... then ... else ...`
–	Iteration	`loop ... end`

Non-Standard Languages

- Lisp, Prolog:
  - No (explicit) assignment
  - No (explicit) goto
  - No (explicit) iteration
- Clos:
  Pass "function" to "object"
  (not *objects* to *function*)

- Need: new view of Programming
  NOT
  "*do* $\boxed{\textit{this}}$ *then do* $\boxed{\textit{that}}$ *then do* $\boxed{\textit{other}}$"

Emphasis of CMPUT325

- On **FEATURES** of Program Paradigms
  Study of Features (not just their *use*)

- *NOT* full languages,
  big systems, documentation, etc.

- "Theoreticians" of Languages
  not (just) "Practicianers"

(Practical details of running ⇒ in Tutorial)

Types of Programming Languages

- ◦ Procedural Programming
  assembly, Fortran, Basic, Cobol, PL/I, Turing, Pascal, Algol, Ada, C, APL, ...
- • Functional (or Applicative) Programming:
  (Common) Lisp
- • Declarative (or Logic) Programming:
  Prolog
- ◦ Object Oriented Programming
  Clos (Common Lisp)
- ◦ Concurrent Programming
  Turing+

Distinguishing Features

- • Lisp [Functional Programming]
  Execution ≈ *function* application
  Recursion does the work
  Data structures include programs
- • Prolog [Declarative Programming]
  Based on (limited) first-order logic
  Execution ≈ search for proof
  Retrieval by pattern matching
  Programs can be inverted
- ◦ Clos [Object Oriented Programming]
  Pass process to objects
- ◦ Turing+ [Concurrent Programming]
  Concurrency, Synchronization

Uses of Different Languages

- • Lisp:
  mainly AI
  also symbolic math and CAD/CAM
- • Prolog:
  mainly AI
  some database work too
- ◦ Clos:
  AI systems
  Window management systems
  Some operating systems
- ◦ Turing+:
  operating systems