

# Structure of Programming Languages – Lecture 1

CS 636 – 536

April 16, 2009

- 1 About this Course
  - Syllabus
- 2 FORTH
- 3 Homework

## The course goals are to learn:

- That all languages are alike, no matter what disguise they are in.
- How to download, install, and bring up a new language on your own computer.
- How to approach a new language and master it quickly.
- Two languages, FORTH and LISP, that are very different from C, at the introductory level.
- Many things about a variety of other languages, including Python and Ruby.
- The structures that are used to implement compilers and interpreters.
- The benefits and drawbacks of various language designs, including late binding, byte code interpreters, and garbage collection.
- How types, objects, generic functions, and polymorphism are

# The Textbook

## A. Fischer and F. Grodzinsky, The Anatomy of Programming Languages, Prentice-Hall, 1993

- This book is out-of-print, but used copies are often available on the web.
- Use the "Chapters from the text" link, on the website to access PDF copies of the chapters.

Why use such an old book? This is a book you can read; the sentences are short. I had a good laugh when one reviewer said it was "too advanced" for her state college! It explains complicated things in simple language.

# Grades.

- Do all the work = A+
- Do some of every type well and do all of some type well = A
- don't turn in anything = ' : (

## Weekly Written Homework

The goal is to answer a few questions clearly and precisely, in your own words.

Example: define “dog”.

- First effort: A 4-legged animal with a tail that barks. (not bad)
- More precise: A 4-legged animal with a tail that barks. that is of the family canine and typically is domestic.

Another example: Stephen, how far away is your home?

- About a 5 to 7 minute walk.
- More precise: A 7 minute walk at a pace of 17 minutes per mile.

## Weekly Written Homework

Here is a confusing and foggy statement about Java concocted by John:

*Java uses beans.*

Another example, from an end-of-term student evaluation:

*Dr. Eggert is a mad freeze.*

I think it means he is “crazy, man, crazy” and “cool” .

I will be assigning written homework. Not a whole lot – but regularly. When you write answers, strive for clarity and precision.

# The FORTH Game

Hello, I am your friendly intelligent computer. Ask me questions and I will give you a sensible answer. Your goal? Starting from zero, learn as much as you can about FORTH.

- 1 FORTH was designed for small applications and embedded systems.
- 2 FORTH has been standardized and restandardized several times. I am most familiar with FORTH83. The differences from version to version are slight and often relate to the implementation, not the language.
- 3 Today FORTH is used by almost everyone all the time – pdf is based on it.

# Strengths of FORTH.

Why Learn FORTH? Why does this language exist?

- 4 It has a small compiler. You can port it to a new system easily.
- 5 It has a small run-time system. Great for embedded systems.
- 6 It is almost as fast as assembly code (about a 10% penalty).
- 7 A student can study it and learn how the entire translator works.
- 8 By studying FORTH, you learn how other languages are implemented.

## The FORTH Game - 3

- 9 FORTH uses a byte-code compiler. Byte codes are an architecture-independent way to write a program. They are too low-level for humans and easy to translate into machine instructions. The source code is compiled into this intermediate form, then interpreted.
- 10 What IDE will we be using?  
FORTH, like many languages of its class, is a self-contained system. In addition to the compiler and interpreter, every FORTH system supplies an editor and an assembler.
- 11 However, my advice is to use your favorite editor to write programs and store them in normal files, called xxxx.for When you want to bring a piece of program into the FORTH system, you can either LOAD a file or use the mouse to paste some lines into the window.

## The FORTH Game - 4

- 12 Describe the FORTH syntax.
  - Space delimited, free form, with a small number of keywords.
- 13 Is FORTH case-INsensitive?
  - Yes.
- 14 Can we define local variables?
  - No, all variables are global.
- 15 Please define a variable:    `variable myvar`

## The FORTH Game - 5

- 17 What data types are supported?
  - All numbers are integers.
  - For example, `/` is integer division: integer result, int remainder. `20 3 /` is 6.
- 18 You can emulate fixed point numbers pretty easily. Hooks are built in.
- 19 Some operations are built in for single-byte values (characters).
- 20 Most strings in FORTH are literal strings. If you really wanted to allocate space for a variable string, you could.

## The FORTH Game - 6

- 21 FORTH consistently uses postfix notation, even for control structures.
- 22 All computation is stack based.
- 23 How do we modify a variable? – Just like in C.  
First, get the data onto the run-time stack,  
then apply an operation to it,  
then (possibly) store the result in a variable.  
`myvar @ 1 + myvar !` ( add 1 to myvar )
- 24 @ is “fetch”: it copies the value of a variable onto the stack.
- 25 ! is “bang”: it stores the value given (second to top of stack) into the address given (top of stack).

# FORTH Basics

There are several classes of operations (see reference on website):

- 28 Arithmetic: + - \* / mod /mod
- 29 Stack-manipulation: dup drop rot 2dup swap, others.
- 30 Function definition : : ; [ ]
- 31 Output: . ." .S
- 32 Control: if loop else while begin repeat

FORTH information and examples from the first lecture are continued in the document named "April6b.pdf".

# Homework 1: Starting with FORTH

Download and install a FORTH system on your computer. The course website contains advice about where to go to download. Run the gcd program (with its test data) on your system and make sure everything works properly. Turn in a transcript of your run.