

# Computing Science (CMPUT) 325

## Nonprocedural Programming

Martin Müller

Department of Computing Science  
University of Alberta

Winter 2018

# Functional Programming Languages

- Many dozens of functional languages exist
- Lisp is one of the oldest, and branched into many dialects
- We use Common Lisp which has an ANSI standard

# History of Functional Programming

- Lambda calculus: developed by Alonzo Church, mathematician and logician, in 1930s
  - You will meet him again, e.g. Church-Turing thesis, Church-Rosser Theorem
- Formal system for expressing computation
- Main concepts: function abstraction, function application, variable binding, substitution
- Equivalent in expressive power to Turing machines
- A minimalistic, theoretical basis for functional programming
- We will study it later in this part of the course

# LISP History

- LISP: stands for LISt Processor
- Developed by John McCarthy in 1958
- “second-oldest language in widespread use today”
- Common Lisp: current ANSI standard from 1994, revised 2004
- Many important dialects - I will review a few
- Big general purpose language, we focus on the “pure functional” subset

# Scheme

- Major dialect of Lisp, different from Common Lisp
- Developed in the 70's by Steele and Sussman at MIT
- Also has official IEEE standard, but many variants exist
- Used in many schools, sometimes as first programming language
- Used e.g. in GIMP graphics editor, Final Fantasy, Google App Inventor
- Often used as a scripting language for extending programs
- Implementations e.g. MIT/GNU Scheme, Stalin compiler, Kawa compiler to JVM

# Clojure

- Lisp dialect, based on Common Lisp
- Focus on multithreaded programming, immutable data
- Runs on the Java virtual machine (JVM)
- Adds a **type system**
- Used in production code by large companies such as Walmart
- See `clojure.org`

# Emacs Lisp

- By Richard Stallman (of FSF, GNU, gcc fame)
- Used in GNU emacs and XEmacs editors
  - Small core with primitives written in C
  - Everything else is in Lisp
- Used to customize and extend emacs
- Also used as a scripting language, e.g. directly from command line

## Hy (or Hylang)

- Lisp dialect, integrated with Python
- “Lisp front end to Python’s abstract syntax tree (AST)”
- Lisp data compiled into AST
- Can use Python libraries with Lisp in this way
- See [hylang.org](http://hylang.org)



## Some Other Functional Languages

- Haskell
- ML, Caml, OCaml and F#
- APL
- Traditional languages with functional programming elements

# Haskell

- Pure functional language - no side effects
- Strong static typing (unlike Lisp)
- Lazy evaluation
- GHC - open source interpreter and compiler
- See `haskell.org`

# ML, Caml, OCaml and F#

- “Impure” functional language, allows side effects
- Eager evaluation - evaluates subexpressions
- ML developed from 1970s
- F# by Microsoft, part of CLI, .NET and Visual Studio, strongly typed
- Caml, OCaml based on ML, popular in theorem proving community

## APL and J

- Developed by Ken Iverson from 1960s
  - Iverson is from Camrose, Alberta and won the Turing Award in 1979
- Very concise language based on functions and **operators** (like higher-order functions)
- Central data type is array and higher-dimensional arrays (matrices etc)
- Very powerful built-in functions
- Special character set, most functions are a single symbol

# Traditional languages with functional programming elements

- C++11, C++14
- Java 8
- Javascript, e.g. functions as first class objects, lambda functions
- Python
- There are recent books on “Functional Programming using  $x$ ” for all programming languages  $x$  above...

# Summary

- Brief Overview of functional programming (FP) languages
- Lambda calculus as formal basis
- History of Lisp and dialects
- Some other popular and/or influential functional languages
- Recent trend: add FP features to other languages