

Functional Programming



Big Picture

- What we've learned so far:
 - Imperative Programming Languages
 - Variables, binding, scoping, reference environment, etc
- What's next:
 - Functional Programming Languages
 - Semantics of Programming Languages
 - Control Flow
 - Data Types
 - Logic Programming
 - Subroutines, Procedures: Control Abstraction
 - Data Abstraction and Object-Oriented

Organization of Programming Languages, Ching (Fall 2004)



Design of Programming Languages

- Design of imperative languages is based directly on the von Neumann architecture
 - Efficiency is the primary concern, rather than the suitability of the language for software development
- Other designs:
 - Functional programming languages
 - Based on mathematical functions
 - Logic programming languages
 - Based on formal logic (specifically, predicate calculus)

Organization of Programming Languages, Ching (Fall 2004)



Design View of a Program

- If we ignore the details of computation ("how" of computation) and focus on the result being computed ("what" of computation)
- A program becomes a "black box" for obtaining output from input

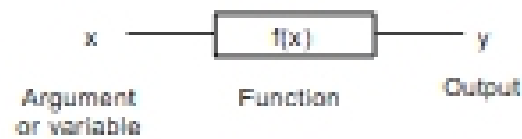


Organization of Programming Languages, Cheney (Fall 2004)



Functional Programming Language

- Functional programming is a style of programming in which the primary method of computation is the application of functions to arguments



Organization of Programming Languages, Cheney (Fall 2004)



Historical Background

1940s:



Alonzo Church and Haskell Curry developed the *lambda calculus*, a simple but powerful mathematical theory of functions.

Organization of Programming Languages, Cheney (Fall 2004)

8



Historical Background

1960s:



John McCarthy developed **Lisp**, the first functional language. Some influences from the lambda calculus, but still retained variable assignments.

Organization of Programming Languages, Chong (Fall 2004)

7



Historical Background

1978:



John Backus publishes award winning article on **FP**, a functional language that emphasizes *higher-order functions* and *calculating with programs*.

Organization of Programming Languages, Chong (Fall 2004)

8



Historical Background

Mid 1970s:



Robin Milner develops **ML**, the first of the modern functional languages, which introduced *type inference* and *polymorphic types*.

Organization of Programming Languages, Chong (Fall 2004)

9
