Two Computational Paradigms:

1. Imperative P.L.s
2. Functional P. L’s

“Von Neuman Architecture”

“Stored Program”

Concept

Two

* R1 – Model of Computation
  + Parallel Prog.
  + Non-Deterministic Prog.
* R2 – Hardware

Programming Language:

* A notation for describing DATA of Instructions

Program:

* A well-defined sentence in a P.L

Binary Strings

* Data
* Instruction.

Program: A finitary specification of a computation

P.L.: A notation for writing a program.

Z+C = {f(x) | x is in N} (Set Builders)

Z+ C = {0,2,3,…}

Language Classification

* Low-Level
  + Machine L
* High-Level
  + Imperative
  + Functional
  + Logic

Language Classification by features

* Sequential
* Concurrent
* Modular
* Parallel
* Distributed
* Object Oriented

Language History

* Imperative
  + FORTRAN
    - Algol-GO
      * Pascal
        + Module
        + Ada
        + CLU
      * Simula
    - BCPL
      * C
        + C++
      * PLI
  + COBOL
* Functional P.L
  + LISP

Language Design

* Regularity
* Generality
* Orthogonality
* Informality

“PRINCIPLE LEAST ASTONISHMENT”