# CSC326 Programming Languages (Lec 1)

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| 1.0 | 2011-09 | | JZ |

# Contents

# 1 Agenda

- Course Objective
- History of Programming Languages
- Paradigm of Programming Languages
- Course Methodology

# 2 Goal

- Paradigms
  - Aware
  - Use
  - Appreciate: Why use
  - Criticize: Why not use
- Batteries
  - Strength of PL decided by its ecosystem
  - Master key libraries
  - Learn how to search, mine and use new libraries
- Web
  - Software are migrating to web/cloud
  - Success depends on how fast one could code: productivity
  - Drive the learning by web programming
    1. Case studies
    2. Projects

# 3 What's PL

A language is intended for use by a person to express a process by which a computer can solve a problem.

a set of conventions for communication an algorithm.

The art of programming is the art of organizing complexity.

# 4 PL History

- Machine language
  - register transfers
  - encoded in binary
- Assembly language
  - one to one mapping to macine language
  - mnemonic for readibility

The main idea is to treat a program as a piece of literature, addressed to human being rather than to a computer.

- Plethora of languages
  - Way too many, and being invented each day
  - Few survived
  - Few is still many

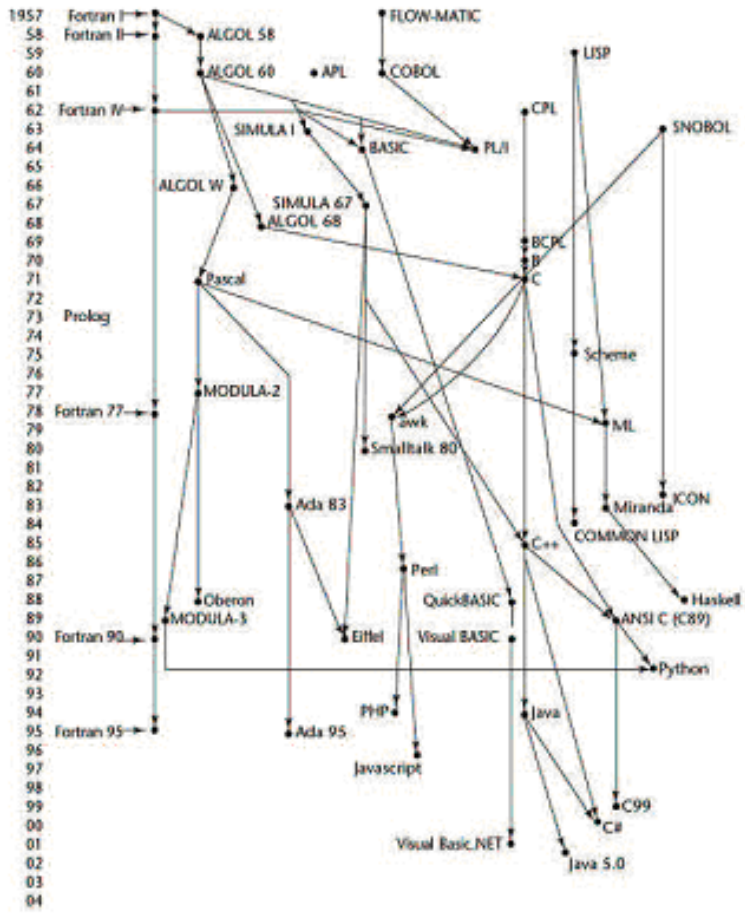


**Figure 2.1** Genealogy of common high-level programming languages from Sebesta, Concepts of programming languages

- Traits of successful languages
  - Intuitive
  - Expressive
  - Efficient
  - Wide deployment (free helps)

# 5 PL Paradigms

## 5.1 Imperiative

- model of Von Nuemann machine

  - Store (memory)
  - ALU
  - Load/Store into memory
- Language constructs
  - variable: states (store)
  - statement: state transitions
  - expressions: computation
- Examples

Fortran

```
    SUM = 0
    DO 11 K=1,N
    SUM = SUM + 2*K
11  CONTINUE
```

C

```
  sum = 0;
  for( k = 1; k <= n; k ++ ) {
      sum += 2*k;
```

Pascal

```
  sum := 0;
  for k := 1 to n do
      sum := sum + 2*k;
```

## 5.2 Functional

- Model of Dataflow machine
  - No store
- Language constructs
  - No statement
  - iteration through recursion
  - Function application

Scheme

```
        (define (sum n)
                        (if (= n 0)
                                0
                                (+ (* n 2) (sum (- n 1)))
                        )
        )
```

### 5.3 And Many More

- Object-based/oriented
- Aspect-oriented
- Event-based
- Message-passing
- Concurrent
- Parallel

## 6 Course Methodology

- Use a single, but multiparadigm language
  - Python
  - Very popular
  - Well designed

```
TIOBE Programming Community Index

. Java
. C
. C++
. C#
. PHP
. Objective-C
. (Visual) Basic
. Python
. Perl
. JavaScript
```

- Content
  - Imperative paradigm
  - OO programming paradigm
  - Meta programming (aspect paradigm)
  - Functional programming paradigm
  - Concurrent programming paradigm
- Case studies
  - With emphasis on web
  - Batteries for web programming
  - Prepare you for an ambitious project
    1. How google works
    2. How twitter works
    3. And more
- Project
  - Step 1: Watch "the social network"

  - Step 2: Brainstorming ideas on web service
  - Step 3: Submit proposal to TA
  - Step 4: Architect
  - Step 5: Do it
  - Step 6: Demo
  - Step 7: Next Facebook?
- Two awards:
  - CSC326 Best Idea Award
  - CSC326 Best Code Award

## 7 Administration

- Web: http://www.eecg.toronto.edu/~jzhu/csc326
- Forum:
- Tutorial: — Biweekley — Problem set and project
- Grade
  - Final: 40%
  - Project: 20%
  - Midterm: 20%
  - Homework:20% (4)
  - Bonus: 10%