



Programming Languages

Contact Information

- Adelina Aleksieva-Petrova
- Email: aaleksieva@tu-sofia.bg
- Room: 3311A
- Office Hours:
On Tuesday – 13:30 – 14:30h

Programming Languages

- The main themes of programming language design and use:
 - **Model of computation**
 - **Expressiveness**
 - types and their operations
 - control structures
 - abstraction mechanisms
 - tools for programming in the large
 - **Ease of use: Writeability / Readability / Maintainability**

Models of Computation

- **Imperative:** programs have mutable storage (state) modified by assignments
 - by far the most common and familiar
- **Functional** (applicative): programs are pure functions
 - much use in AI, formal semantics, language research
- **Declarative:** programs are unordered sets of assertions and rules
 - Prolog, data base applications

The Generations of Programming Languages (1/2)

- First-generation languages (1954–1958)
 - FORTRAN I Mathematical expressions
 - ALGOL 58 Mathematical expressions
 - Flowmatic Mathematical expressions
 - IPL V Mathematical expressions
- Second-generation languages (1959–1961)
 - FORTRAN II Subroutines, separate compilation
 - ALGOL 60 Block structure, data types
 - COBOL Data description, file handling
 - Lisp List processing, pointers, garbage collection

The Generations of Programming Languages (2/2)

- Third-generation languages (1962–1970)
 - PL/I FORTRAN + ALGOL + COBOL
 - ALGOL 68 Rigorous successor to ALGOL 60
 - Pascal Simple successor to ALGOL 60
 - Simula Classes, data abstraction
- The generation gap (1970–1980).

Object-orientation boom

- 1980–1990, but few languages survive
 - Smalltalk 80 Pure object-oriented language
 - C++ Derived from C and Simula
 - Ada83 Strong typing; heavy Pascal influence
 - Eiffel Derived from Ada and Simula

Emergence of frameworks (1990–today)

- Visual Basic - eased development of the graphical user interface (GUI) for Windows applications
- Java - successor to Oak; designed for portability
- Python - object-oriented scripting language
- J2EE - Java-based framework for enterprise computing
- .NET - Microsoft's object-based framework
- Visual C# - Java competitor for the Microsoft .NET Framework
- Visual Basic .NET - Visual Basic for the Microsoft .NET Framework

Common Ideas

- Modern imperative languages (Ada, C++, Java) have similar characteristics:
 - large number of features (grammar with several hundred productions, 500 page reference manuals...)
 - a rich type system
 - procedural mechanisms
 - object-oriented facilities
 - abstraction mechanisms, with information hiding
 - several storage-allocation mechanisms
 - facilities for concurrent programming
 - facilities for generic programming

Predictable performance vs. ease of writing

- Low-level languages mirror the physical machine:
 - Assembly, C, Fortran
- High-level languages model an abstract machine with useful capabilities:
 - ML, Setl, Prolog, Python
- Wide-spectrum languages try to do both, more or less well:
 - Ada, C++, Java
- High-level languages are often interpreted, have garbage collector. Cost of operations is not directly visible.
 - Java is a hybrid

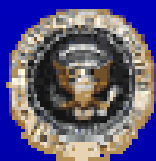
Language as a tool for thought (Iverson)

- Drawing a histogram in APL:
 - “*“ $[V \circ \geq 1 \lceil V]$
 - Is it natural ? (only if you happen to think that way)
 - **Role of language as a communication vehicle among programmers is more important than ease of writing**
 - APL is an extreme case (write-only language)
 - All languages have the same expressive power; arguments of the form “you can’t do this in X” are meaningless.
 - But.. Idioms in language A may be useful inspiration when writing in language B.

Properties

- need to be precise
- need to be concise
- need to be expressive
- need to be at a high-level (lot of abstractions)

High-level Abstract Description to Low-level Implementation Details



President



My poll ratings are low,
lets invade a small nation



General



Cross the river and take
defensive positions



Sergeant



Forward march, turn left
Stop!, Shoot



Foot Soldier



TIOBE Programming Community Index for September 2011

Position Sep 2012	Position Sep 2011	Delta in Position	Programming Language	Ratings Sep 2012	Delta Sep 2011	Status
1	2	↑	C	19.295%	+1.29%	A
2	1	↓	Java	16.267%	-2.49%	A
3	6	↑↑↑	Objective-C	9.770%	+3.61%	A
4	3	↓	C++	9.147%	+0.30%	A
5	4	↓	C#	6.596%	-0.22%	A
6	5	↓	PHP	5.614%	-0.98%	A
7	7	=	(Visual) Basic	5.528%	+1.11%	A
8	8	=	Python	3.861%	-0.14%	A
9	9	=	Perl	2.267%	-0.20%	A
10	11	↑	Ruby	1.724%	+0.29%	A
11	10	↓	JavaScript	1.328%	-0.14%	A
12	12	=	Delphi/Object Pascal	0.993%	-0.32%	A
13	14	↑	Lisp	0.969%	-0.07%	A
14	15	↑	Transact-SQL	0.875%	+0.02%	A
15	39	↑↑↑↑↑↑↑↑↑↑	Visual Basic .NET	0.840%	+0.53%	A
16	16	=	Pascal	0.830%	-0.02%	A

TIOBE Programming Community Index

