Part IB, Part II(General) and Diploma

Comparative Programming Languages

by

Martin Richards

mr@cl.cam.ac.uk

http://www.cl.cam.ac.uk/users/mr/

University Computer Laboratory
New Museum Site
Pembroke Street
Cambridge, CB2 3QG

The Course

- This a brand new course.
- It will cover language design issues of a variety of programming languages.
- It will give thumb nail sketches of several languages.
- Some languages, particularly C and C++, will be covered in slightly nore detail, since any computer scientist should be able to read code in these two.

Books

- Pratt, T.W and Zelkowitz M.V. *Programming Languages – Design and Implementation*, Prentice Hall (1995)
- Banahan, M. *The C Book*, Addison Wesley
- Stroustrup, B. *The C++ Programming Language*
- Others

Why Study Programming Languages?

- To improve your ability to develop effective algorithms.
- To improve your use of your existing language.
- To increase your vocabulary of useful programming constructs.
- To allow a better choice of programming language.
- To make it easier to learn a new language.
- To make it easier to design a new language.

History

1951-55: Experimental use of expression compilers.

1956-60: FORTRAN, ALGOL 60, COBOL, LISP

1961-65: APL notation, ALGOL 60 (revised), SNOBOL, CPL

1966-70: APL, SNOBOL 4, FORTRAN 66, SNOBOL 4, BASIC, SIMULA 67, ALGOL 68, ALGOL-W, BCPL

History

1971-75: Pascal, PL/1 (Standard), C, Scheme, Prolog

1976-80: Smalltalk, Ada, FORTRAN 77, ML

1981-85: Smalltalk-80, Growth of Prolog, Ada 83

1986-90: C++, SML

1991-95: Ada 95, TCL, Perl

1996-00: Java

Changing Influences

- Computer capabilities.
- Applications.
- Programming methods.
- Implementation methods
- Theoretical studies.
- Standardisation.

What makes a good language

- Clarity, simplicity, and unity.
- Orthogonality.
- Naturalness for the application.
- Support of abstraction.
- Ease of program verification.
- Programming environments.

What makes a good language

- Portability of programs.
- Cost of use.
  - Cost of execution.
  - Cost of program translation.
  - Cost of program creation, testing, and use.
  - Cost of program maintenance.

Application Domains

- Business processing.
- Scientific.
- System.
- AI.
- Publishing.
- Process control
- Embedded systems.
- New paradigms.