

CS 460

Programming Languages

Fall 2008

Dr. Watts

(Week 1)



Course Administration

- Survey
- Syllabus
- Schedule
- Course Manual

- BASIC
- FORTRAN
- Pascal
- COBOL
- BPL
- Audit Reporter
- RPG
- JCL
- SNOBOL
- APL
- ALGOL
- BAL
- SAS
- SPSS
- Ada
- LISP
- C
- Logo
- QBasic
- C++
- MFC
- HTML
- Scheme
- Java
- Action Script



Why do we study Programming Languages?



- Choosing languages
- Learning languages
- Efficient program implementation
- Designing and implementing new languages
- Expressing ideas
- Overall understanding

Influences on Language Design



- Architectures
- Domains
- Paradigms



Programming Domains

- Science and Mathematics
 - FORTRAN – FORMula TRANslator
- Business
 - COBOL – Common Business Oriented Language
- Education
 - BASIC – Beginners All-purpose Symbolic Instruction Code
- Artificial Intelligence
 - LISP
- Systems
 - Assembly languages, C



Programming Paradigms

- Procedural
 - FORTRAN, COBOL, BASIC, Pascal
- Functional
 - LISP, Scheme
- Logical
 - Prolog
- Object Oriented
 - Smalltalk, Java
- Interactive
 - Java
- Scripting
 - RPG, Java Script
- Web
 - HTML, XML
- Hybrid
 - C++

Language Design Factors



- Readability
- Simplicity
- Orthogonality
- Control Structures
- Data Types/Structures
- Writability
- Reliability
- Cost

Implementation



- Importance of Standards
- Compilation vs interpretation vs hybrid
- Components of language translation
 - Lexical analysis
 - Syntactical analysis
 - Semantic analysis
 - Intermediate code generation
 - Optimization
 - Target code generation
- Portability

Programming Environments



- Stand alone tools
- Integrated Development Environments
 - IDEs
 - Turbo Pascal (Borland)
 - MS Visual C++
 - Code Warrior
 - MS .net

History of Programming Languages



- 5+ generations of computing history
 - Dates
 - Signature Hardware
 - People
 - Companies
 - Languages
 - Milestones

Generation 0



- Before 1940
- Mechanical
 - Abacus
 - Slide Rule
 - Analytic and Difference Engines
 - Punch card
- Charles Babbage, Lady Ada Lovelace, Blaise Pascal, Herman Hollerith
- “Programming by Screwdriver”
 - Dials, other mechanical devices
 - Concepts of subprograms and loops

Generation 1

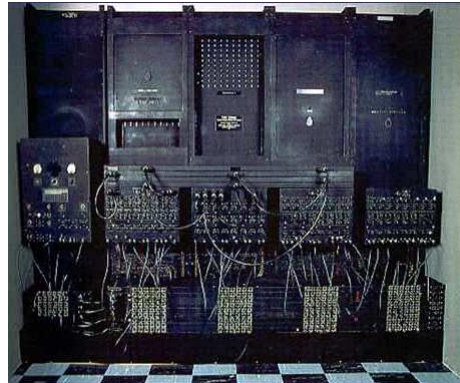


- 1940's
- Electro-mechanical
 - Relays
 - Vacuum tubes
- Alan Turing, Presper Eckert & John Mauchley, John V. Atanasoff, John VonNeuman, Konrad Zuse
- American Tabulating Company (later IBM)
- Machine Language
 - Plankakül

Generation 1 (continued)



- Turing machine
- Enigma Code
- ENIAC
- Stored program design



Generation 2



- 1950's
- Transistors
- IBM, Univac
- Assembly Languages
 - Short code
- Movement of computing from scientific laboratory to real world



Generation 3

- 1960's – 1980's
- Integrated Circuits
- Bill Gates, Steve Wozniak & Steve Jobs
- Microsoft, Wang, IBM, Apple
- High Level Language
 - FORTRAN, COBOL, BASIC
 - LISP, Smalltalk, Prolog
 - Algol, Pascal
 - Ada



Generation 3 (continued)

- First Compiler
- First Interpreter
- Algol 60 Specification
 - BNF – Bachus Naur Form (grammar spcification)
 - Standardization
- Structured Programming
 - Nested Control Structures
 - Case Statement
 - Single Pass Compilation
 - Pascal

Generation 3 (continued)



- Data Structures
 - Arrays, Lists
 - Strings
 - Records
- Multi-processing Operating Systems
 - C / UNIX
- Multi-user Operating Systems (time sharing)
- Personal Computer
 - Microprocessor
 - BASIC
 - Apple, IBM PC & Microsoft DOS
 - *Triumph of the Nerds: The Rise of Accidental Empires*

Generation 3 (continued)



- Dynamic Language
 - APL, SNOBOL
- All-purpose Languages
 - PL/I – all current data structures
 - Ada – Department of Defense
 - Software Engineering
 - Before its time – slow compiler; slow development
- Object-Oriented Programming
 - Smalltalk, Eiffel, C++, Java

Generation 4



- 1980's – 2000's
- VLSI – very large scale integrated circuits
- Distributed and Parallel Processing
- World Wide Web
- GUI Emergence – high powered graphics
 - Java, Visual BASIC, C#
- Special Purpose Languages
 - HTML, Java/Swing, Action Script
- Scripting Languages
 - Java Script, Python, PHP, Ruby

Generation 5



- Still to come
- Post silicon
- Natural Languages
- AI

Thoughts on Computer Programming Languages



- The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offence. (Edsger Dijkstra)
- Consistently separating words by spaces became a general custom about the tenth century A.D., and lasted until about 1957, when FORTRAN abandoned the practice. (Sun FORTRAN Reference Manual)
- Cobol has almost no fervent enthusiasts. As a programming tool, it has roughly the sex appeal of a wrench. (Charles Petzold)
- C++ is the only current language making COBOL look good. (Bertrand Meyer)
- C++ has its place in the history of programming languages. Just as Caligula has his place in the history of the Roman Empire. (Robert Firth)
- Arguing that Java is better than C++ is like arguing that grasshoppers taste better than tree bark. (Thant Tesson)
- Java is, in many ways, C++-. (Michael Feldman)
- If Java had true garbage collection, most programs would delete themselves upon execution. (Robert Sewell)
- It is practically impossible to teach good programming style to students that have had prior exposure to BASIC; as potential programmers they are mentally mutilated beyond hope of regeneration. (Edsger Dijkstra)
- In my egotistical opinion, most people's C programs should be indented six feet downward and covered with dirt. (Blair P. Houghton)
- C++ is history repeated as tragedy. Java is history repeated as farce.
- (Scott McKay)
- Unix and C are the ultimate computer viruses. (Richard P Gabriel)