CIT 103 & CIT 104

Object Oriented Programming

By

Md. Palash Uddin

Lecturer

Dept. of CIT

Hajee Mohammad Danesh Science and Technology University, Dinajpur.

Lecture Objective

- To know the revolution of computer
- To know the revolution of programming language
- To provide inspiration for invention and innovation

What is a Programming Language?

- A tool for instructing machines
- A notation for algorithms
- A means for communication among programmers
- A tool for experimentation
- A means for controlling computer-controlled gadgets
- A means for controlling computerized devices
- A way of expressing relationships among concepts
- A means for expressing high-level designs
- All of the above!
  - And more

What is a Programming Language?

- A programming language is a tool for developing executable models for a class of problem domains.
- A programming language is a notational system for describing computation in a machine-readable and human-readable form.

— Louden

Early History: John Napier and Blase Pascal

- The first mechanical calculating device is the **abacus** used around **2500 BC** to add and subtract.
- In **1610**, Scottish mathematician **John Napier** developed a series of rods made of bones that could be arranged to generate the **products of selected numbers.**
- In **1865**, French mathematician **Blase Pascal** improved on this concept and produced a mechanical calculator called **Pascaline** perfoming addition and subtraction.
- In **1920**, Thomas de Colmar of France produced the **arithmometer**, the first four-arithmetic function practical mechanical calculator.

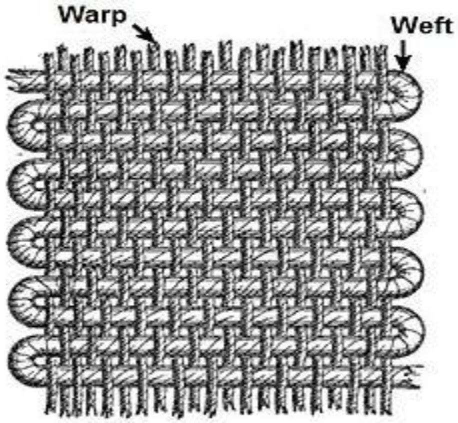
Early History: Johann Helfrich von Müller

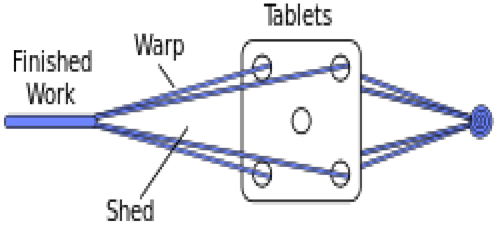
- **Johann Helfrich von Müller** (January 16, 1746 – 1830)
- An engineer in the Hessian army
- Conceived of the idea of a **difference machine**
- This was described in a book **published in 1786**
- But Müller was **unable to obtain funding** to progress with the idea

Early History: Joseph Marie Jacquard

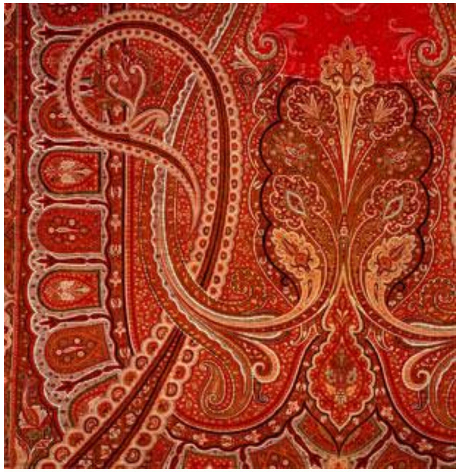
- Joseph Marie Jacquard (1752–1834)
- Jacquard Loom of early 1800s
  - — Translated punch card patterns into cloth designs

Early History: Joseph Marie Jacquard



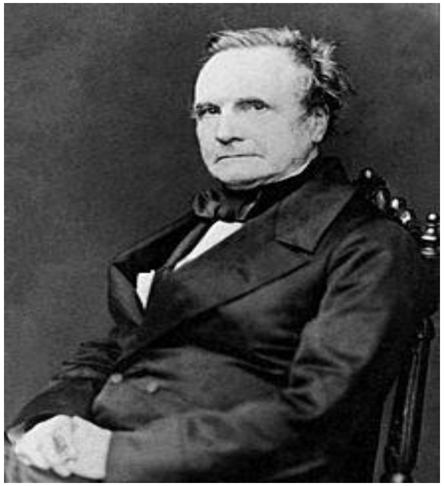


Early History: Joseph Marie Jacquard



Early History: Charles Babbage

- **Charles Babbage** (26 December 1791 – 18 October 1871)
- English Mathematician Philosopher, mechanical engineer
- Invented the first **mechanical computer**
- Considered **"father of the computer”**
- ***Difference engine***
- **Analytical engine**



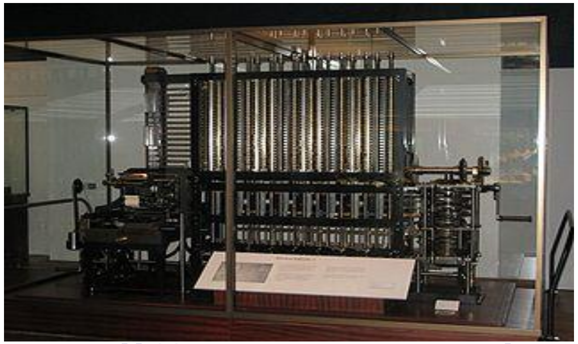
Early History: Charles Babbage *Difference Engine*

- Based on **Johann Helfrich von Müller's Difference machine.**
- **Mechanical calculator** designed to tabulate polynomial functions (**logarithmic and trigonometric functions**)
- **On June 14, 1822**, Charles Babbage proposed the use of such a machine
- **In 1823, the British government** gave Babbage **£17000** to start work on the project
- Although Babbage's design was **technically feasible**, **no one** had built a mechanical device
- By the time the **government killed the project in 1842** because
  - ✓ Babbage had moved on to an **analytical engine**
  - ✓ Babbage had **conflicts with his chief engineer**, ***Joseph Clement***

Early History: Charles Babbage

Difference Engine

- Later an improved **"Difference Engine No. 2"** has been produced between 1847 and 1849
- **Per Georg Scheutz** built several **difference engines from 1855 onwards**, one of which was **sold to the British government in 1859**.
- **Martin Wiberg improved Scheutz's construction** but used his device only for producing and publishing printed logarithmic tables.



The London Science Museum's difference engine

Early History: Charles Babbage *Analytical engine*

- **First mechanical general-purpose computer** described in **1837**
- Arithmetic logic unit, control flow/conditional branching, loops, and integrated memory
- Programs in cards with data and operations directed by **Jacquard loom**
- employed ordinary **base-10** fixed-point arithmetic
- A memory capable of holding 1,000 numbers of 40 decimal digits each (ca. 16.7 kB).
- All four arithmetic operations, comparisons and optionally square roots.
- The programming language to be employed by users was akin to modern day **assembly languages**.

Early History: Charles Babbage *Analytical engine*

- In 1842, the Italian mathematician **Luigi Menabrea** (Babbage had met while travelling in Italy) wrote a description of the engine in French
- In 1843, **Ada Byron or Ada Lovelace** (Countess of Lovelace) translate into English
- She has been described as the **first computer programmer.**
- Late in his life, Babbage sought ways to build a simplified version of the machine, and assembled a small part of it before his death in **1871**.
- In 1878, a committee of the **British Association for the Advancement of Science** recommended against constructing the Analytical Engine.
- In 1910, Babbage's son **Henry Prevost Babbage** reported that a part of the mill and the printing apparatus had been constructed.

Early History: Ada Lovelace

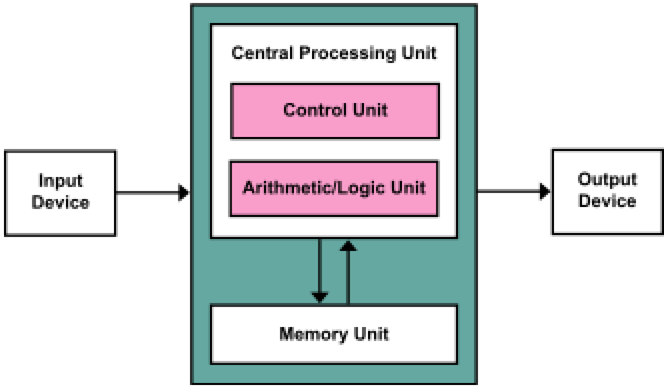
- Ada Lovelace (10 Dec 1815 – 27 Nov 1852)
- **World's first computer programmer**
- *“The engine can arrange and combine its numerical quantities exactly as if they were letters or any other general symbols; And in fact might bring out its results in algebraic notation, were provision made.”*
- The modern computer programming language **Ada** is named in her honor

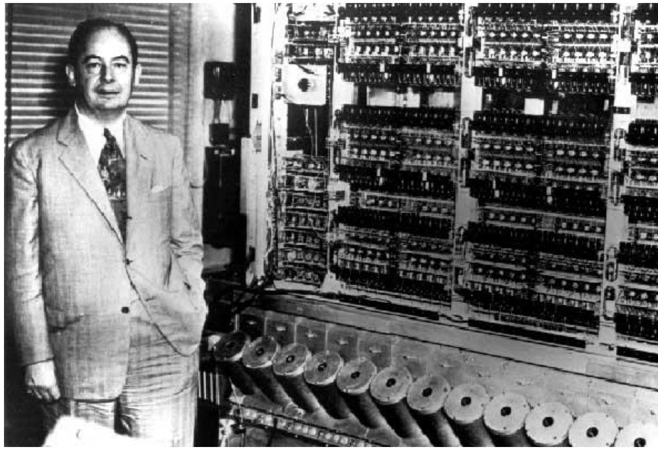
Early History: Konrad Zuse

- Konrad Zuse (22 June 1910 – 18 December 1995)
- Designed **Z3**, an **electromechanical computer**
- The **world's first working programmable**, fully automatic digital computer in 1941
- **lacked of the conditional branch.**
- From 1943-1945, designed **Plankalkül**, first high-level non-von Neumann programming language
- This notation (**never implemented**) has features than can be found in many existing programming languages.

Early History: John von Neumann

- December 28, 1903 – February 8, 1957
- Led a team that built computers with stored programs and a central processor
- Based on **von Neumann architecture**
- **Electronic Numerical Integrator And Computer,** ENIAC first **electronic general-purpose computer** in 1946
- however, was also programmed with patch cords.





Von Neuman with ENIAC

History of Revolution of Modern Computer

Computer History Year/Enter	**Computer History Inventors/Inventions**	**Computer History Description of Event**
1822	Charles Babbage- Difference Engine (Never Built until 1990s)	No Programming
1837	Charles Babbage Analytical Engine (Never Built)	Program-controlled by punched cards
1936	Konrad Zuse **- Z1 Computer (Germany)**	First freely programmable computer.
1942	John Atanasoff & Clifford Berry Atanasoff–Berry Computer ABC (US)	Who was first in the computing biz is not always as easy as ABC.
1944	Howard Aiken & Grace Hopper **Harvard Mark I Computer (US)**	The Harvard Mark 1 computer.
1946	John Presper Eckert & John W. Mauchly **ENIAC 1 Computer (US)**	20,000 vacuum tubes later...
1948	Frederic Williams & Tom Kilburn **Manchester Baby Computer & The Williams Tube**	Baby and the Williams Tube turn on the memories.

History of Revolution of Modern Computer

Computer History Year/Enter	**Computer History Inventors/Inventions**	**Computer History Description of Event**
1947/48	John Bardeen, Walter Brattain & Wiliam Shockley **The Transistor**	No, a transistor is not a computer, but this invention greatly affected the history of computers.
1951	John Presper Eckert & John W. Mauchly **UNIVAC Computer**	First commercial computer & able to pick presidential winners.
1953	International Business Machines **IBM 701 EDPM Computer**	IBM enters into 'The History of Computers'.
1954	John Backus & IBM **FORTRAN Computer Programming Language**	The first successful high level programming language.
1955 (In Use 1959)	Stanford Research Institute, Bank of America, and General Electric **ERMA and MICR**	The first bank industry computer - also MICR (magnetic ink character recognition) for reading checks.

History of Revolution of Modern Computer

Computer History Year/Enter	**Computer History Inventors/Inventions**	**Computer History Description of Event**
1958	Jack Kilby & Robert Noyce **The Integrated Circuit**	Otherwise known as 'The Chip'
1962	Steve Russell & MIT **Spacewar Computer Game**	The first computer game invented.
1964	Douglas Engelbart **Computer Mouse & Windows**	Nicknamed the mouse because the tail came out the end.
1969	**ARPAnet**	The original Internet.
1970	**Intel 1103 Computer Memory**	The world's first available dynamic RAM chip.
1971	Faggin, Hoff & Mazor **Intel 4004 Computer Microprocessor**	The first microprocessor.
1971	Alan Shugart &IBM **The "Floppy" Disk**	Nicknamed the "Floppy" for its flexibility.

History of Revolution of Modern Computer

Computer History Year/Enter	**Computer History Inventors/Inventions**	**Computer History Description of Event**
1973	Robert Metcalfe & Xerox **The Ethernet Computer Networking**	Networking.
1974/75	**Scelbi & Mark-8 Altair & IBM 5100 Computers**	The first consumer computers.
1976/77	**Apple I, II & TRS-80 & Commodore Pet Computers**	More first consumer computers.
1978	Dan Bricklin & Bob Frankston **VisiCalc Spreadsheet Software**	Any product that pays for itself in two weeks is a surefire winner.
1979	Seymour Rubenstein & Rob Barnaby **WordStar Software**	Word Processors.
1981	IBM **The IBM PC - Home Computer**	From an "Acorn" grows a personal computer revolution
1981	Microsoft **MS-DOS Computer Operating System**	From "Quick And Dirty" comes the operating system of the century.
1983	**Apple Lisa Computer**	The first home computer with a GUI, graphical user interface.
1984	**Apple Macintosh Computer**	The more affordable home computer with a GUI.
1985	**Microsoft Windows**	Microsoft begins the friendly war with Apple.

History of Revolution of Modern Computer

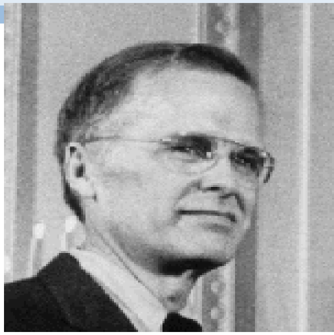
http://inventors.about.com/library/blcoindex.htm

http://www.computerhistory.org/timeline/?category=cmptr

A Brief Chronology

Early 1950s		*“order codes” (primitive assemblers)*
1957	FORTRAN	*the first high-level programming language*
1958	ALGOL	*the first modern, imperative language*
1960	LISP, COBOL	*Interactive programming; business programming*
1962	APL, SIMULA	*the birth of OOP (SIMULA)*
1964	BASIC, PL/I	
1966	ISWIM	*first modern functional language (a proposal)*
1970	Prolog, Pascal	*logic programming is born*
1972	C	*the systems programming language*
1975	Scheme	*teaching languages*
1978	Modula-2, CSP	*Concurrency matures*
1978	FP	*Backus’ proposal*
1983	Smalltalk-80, Ada	*OOP is reinvented*
1984	Standard ML	*FP becomes mainstream (?)*
1986	C++, Eiffel	*OOP is reinvented (again)*
1988	CLOS, Oberon, Mathematica	
1990	Haskell	*FP is reinvented*
1990s	Perl, Python, Ruby, JavaScript	*Scripting languages become mainstream*
1995	Java	*OOP is reinvented for the internet*
2000	C#	

FORTRAN



- **FOR**mula **TRAN**slator (1953-57) arrived for **IBM 70x**
- Developed at IBM under the guidance of **John Backus**
- *John Backus (1953) sought to write programs in conventional mathematical notation, and generate code comparable to good assembly programs*
- continued to evolve, adding new features & concepts.
- FORTRAN II, FORTRAN IV, FORTRAN 66, FORTRAN 77, FORTRAN 90
- **FORTRAN I released April 1957; working by April 1958**
- The current standards : **FORTRAN 2003, FORTRAN 2008**
- Still popular, e.g. for **supercomputers**
- **Backus-Naur Form (BNF)**, a standard notation to describe the syntax of a high level programming language

FORTRAN

Innovations

- Symbolic notation for subroutines and functions
- Assignments to variables of complex expressions
- DO loops
- Comments
- Input/output formats
- Machine-independence

Successes

- Easy to learn; high level
- Promoted by IBM; addressed large user base
- scientific computing

FORTRAN

```
      PROGRAM HELLO
      DO 10, I=1,10
      PRINT *,'Hello World'
10    CONTINUE
      STOP
      END
```

All examples from the ACM "Hello World" project:

www2.latech.edu/~acm/HelloWorld.shtml

ALGOL

History

- **ALGO**rithmic **L**anguage
- Designed by *Committee of PL experts (*international team*) formed in 1955*
- **First version (ALGOL 58) never implemented; criticisms led to ALGOL 60 and ALGOL 60**

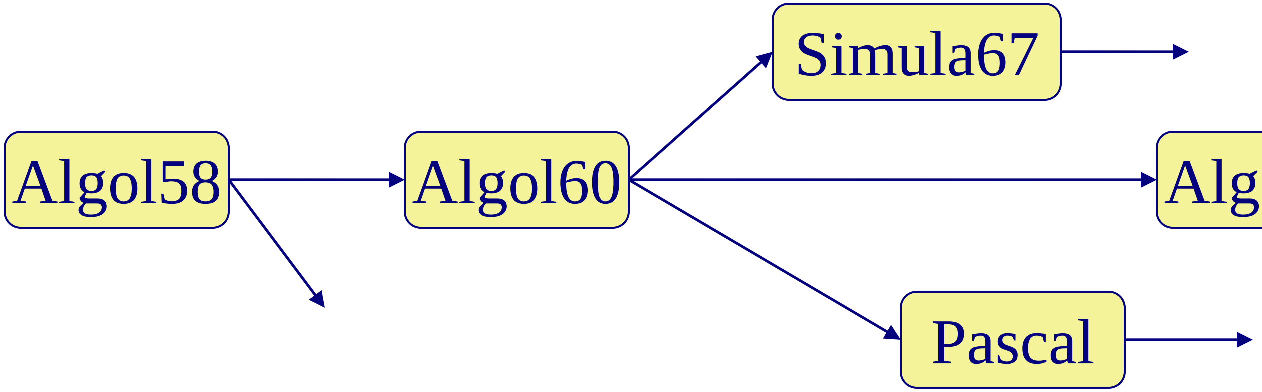
Innovations

- *Universal, and machine-independent*
- BNF (Backus-Naur Form) introduced to define syntax (led to syntax-directed compilers)
- **First block-structured language**; variables with local scope
- Structured control statements
- Recursive procedures
- Variable size arrays
- Semicolon as a statement separator
- Assignment operator was :=

Successes

- **Highly influenced design of other PLs but never displaced FORTRAN**
- **Had strong influence on subsequent languages, especially Pascal, C, and Ada**

ALGOL



Algol58

Algol60

Simula67

Algol68

Pascal

“Hello World” in ALGOL

```
BEGIN
FILE F (KIND=REMOTE);
EBCDIC ARRAY E [0:11];
REPLACE E BY "HELLO WORLD!";
WHILE TRUE DO
      BEGIN
      WRITE (F, *, E);
      END;
END.
```

COBOL

History

- **CO**mmon **B**usiness-**O**riented **L**anguage)
- **US Dept of Defense (DOD) wanted**
- **CODASYL** committee (Conference on Data Systems Languages) Result was COBOL in **1960**
- **Grace Hopper** was involved in development and wrote 1st compiler

Innovations

- Separate descriptions of environment, data, and processes
- Designed to be machine independent, unlike FORTRAN
- Influenced by Fortran, ALGOL58, and English.
- introduced Hierarchical data structures (records) for the first time
- Targeted business applications
- Intended to be readable by managers (!)

Successes

- Adopted as de facto standard by US DOD
- Stable standard for 25 years
- Still *the most widely used PL* for business applications (!)

COBOL

- Continuous evolution: 60, 61, 65, 68, 74, 85, 02
- Used to be very popular in business and government, much less at universities.
- Example:

```
Multiply A by B giving C
Perform <loop body>
    Varying J from 2 by 1
    Until J > N.
```

COBOL

- Dr. Grace Murray Hopper (1906-1992), Rear Admiral-US Navy
- a remarkable woman who grandly rose to the challenges of programming the first computers.
- During her lifetime as a **leader in the field of software development concepts**, she contributed to the transition from primitive programming techniques to the use of **sophisticated compilers**.
- She believed that ‘we've always done it that way’ was not necessarily a good reason to continue to do so.”

“Hello World” in COBOL

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID.     HELLOWORLD.
000300 DATE-WRITTEN.   02/05/96        21:04.
000400* AUTHOR BRIAN COLLINS
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000700 SOURCE-COMPUTER. RM-COBOL.
000800 OBJECT-COMPUTER. RM-COBOL.
001000 DATA DIVISION.
001100 FILE SECTION.
100000 PROCEDURE DIVISION.
100200 MAIN-LOGIC SECTION.
100300 BEGIN.
100400              DISPLAY " " LINE 1 POSITION 1 ERASE EOS.
100500              DISPLAY "HELLO, WORLD." LINE 15 POSITION 10.
100600              STOP RUN.
100700 MAIN-LOGIC-EXIT.
100800              EXIT.
```

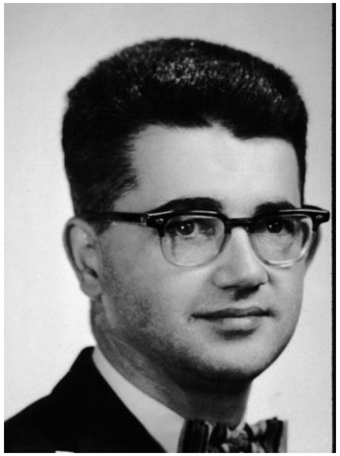
LISP (1960)

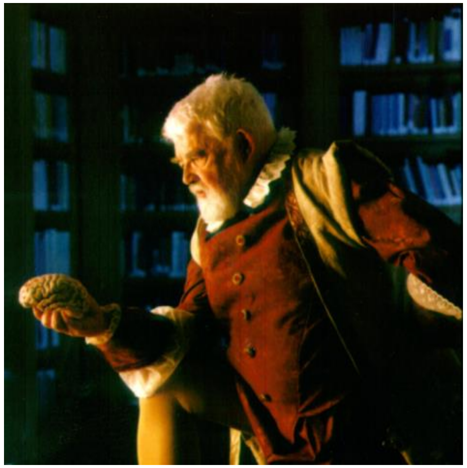
- **LIS**t **P**rocessing language
- Designed at MIT by John McCarthy IBM 704 by
- Artificial Intelligence (*AI) research needed a language that:*
    - Process data in lists (rather than arrays)
    - Handles symbolic computation (rather than numeric)
- *Pioneered* Interactive *Functional programming*
    - No need for variables or assignment
    - Control via recursion and conditional expressions
- Status
    - Still the dominant language for AI
    - COMMON LISP and Scheme are contemporary dialects
    - Meta language (ML), Miranda, and Haskell are related languages

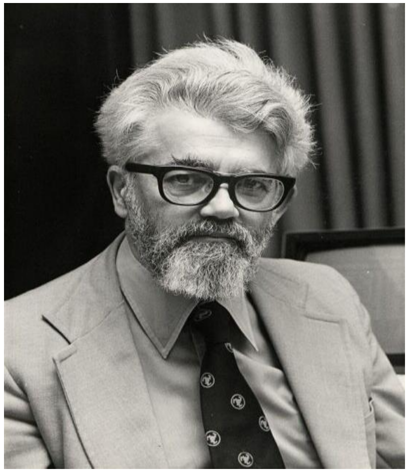
LISP (1960)

John McCarthy (1927-2011)

- Stanford University
- AI pioneer







SIMULA (1962)

History

- **SIMULA**tion programming languages
- Designed and built by **Ole-Johan Dahl** and **Kristen Nygaard** at the Norwegian Computing Centre (NCC) in **Oslo University** between 1962-1967
- An extension of **ALGOL 60** designed for simulation of concurrent processes
- Introduced the central concepts of **OOP: classes and encapsulation**
- Originally designed and implemented as a language for discrete event simulation
- **Predecessor of Smalltalk and C++**
- Now unused
- SIMULA I and SIMULA 67

SIMULA (1962)

Ole-Johan Dahl and **Kristen Nygaard**

PL/I (1964)

History

- Programming Language One in 1964
- A combination of features believed (at the time) best in **FORTRAN, ALGOL 60, COBOL**
- *Designed by committee of* ***IBM*** *and* **SHARE Inc.**
- Intended as (large) general-purpose language for broad classes of applications

Innovations

- Support for concurrency (but not synchronization)
- Exception-handling on conditions
- good for all possible applications

Successes

- Achieved both run-time efficiency and flexibility (at expense of complexity)
- First “complete” general purpose language

“Hello World” in PL/1

```
HELLO:   PROCEDURE OPTIONS (MAIN);

                 /* A PROGRAM TO OUTPUT HELLO WORLD */
                 FLAG = 0;

LOOP:            DO WHILE (FLAG = 0);
                     PUT SKIP DATA('HELLO WORLD!');
                 END LOOP;

END HELLO;
```

BASIC (1964)

- Beginner's All-purpose Symbolic Instructional Code
- Developed at Dartmouth in by **Tom Kurtz, John Kemeny, and a succession of undergraduates**; first ran in 1964
- Intended to introduce students in **non-scientific disciplines to computing**
- Influenced **by FORTRAN and ALGOL**
- Major goal to simplify user interface:
  - Simplicity chosen over efficiency
  - Time sharing over punched cards
  - Distinctions such as int vs real eliminated
  - Automatic defaults for declarations, values, arrays, output format, etc.
  - Clear error messages
  - Students had access to computers at all times
- No universal BASIC standard:
  - **ANSI** (American National Standards Institute) is a minimal standard
  - **True Basic** – Kemeny's company
  - Current popular dialects: **Visual BASIC**

ISWIM Language (1966)

- ***ISWIM (If you See What I Mean)***
- Peter Landin (1966) — paper proposal

Prolog : Logic Programming (1970)

- Developed at the **University of Aix Marseille**, by **Comerauer and Roussel**, with some help from **Kowalski at the University of Edinburgh**
- A very high-level programming language.
- Based on formal logic
- In skilled hands, it is a very strong tool.
- Non-procedural
- Can be summarized as being an intelligent database system that uses an inferencing process to infer the truth of given queries
- Prolog is 12 years younger than LISP

“Hello World” in Prolog

```
hello :- printstring("HELLO WORLD!!!!").

printstring([]).
printstring([H|T]) :- put(H), printstring(T).
```

Pascal (1970)

- Designed by **Niklaus Wirth**
  - (member of ALGOL committee; he proposed a revision known as ALGOL-W in 1965)
- A conceptually simplified and cleaned-up successor of ALGOL 60.
- Pascal first implemented in 1970.
- Named after 17th century French philosopher and mathematician **Blaise Pascal**.
- A *great* language for teaching structured programming. ~ 1980-1995.
- Widely used in academic community
- Interesting features:
  - Case statement
  - Facility for user-defined data types
  - Record structure

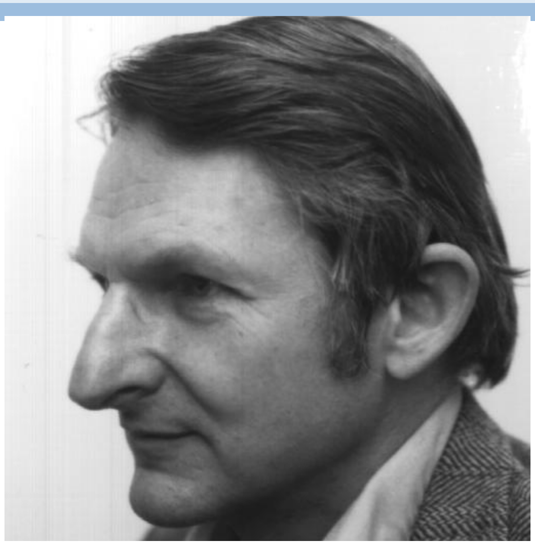
C (1972)

- Designed by **Kenneth Thompson** (born February 4, 1943 invented the **B programming language**, co-invented the **GO programming language**) and **Dennis Ritchie** ((1941-2011)) at Bell Labs in 1972.
- Evolved primarily from B, but also ALGOL 68
- Designed for coding the routines of the **UNIX operating system.**
- **"High level"** systems programming language which created the notion of a portable operating system
- Concise syntax – programs somewhat hard to read, understand, debug, maintain
- No built-in operations for handling composite data types such as strings, sets, and lists.
- Not strongly typed. No run-time type checking. Easily leads to programming errors.
- Provides ability to code low-level operations in a high-level language (Relatively low-level.)
- Once fashionable, still in use, but usually superseded by C++.
- Associated with **Unix** and through that with **Linux** and the **open source movement**

C

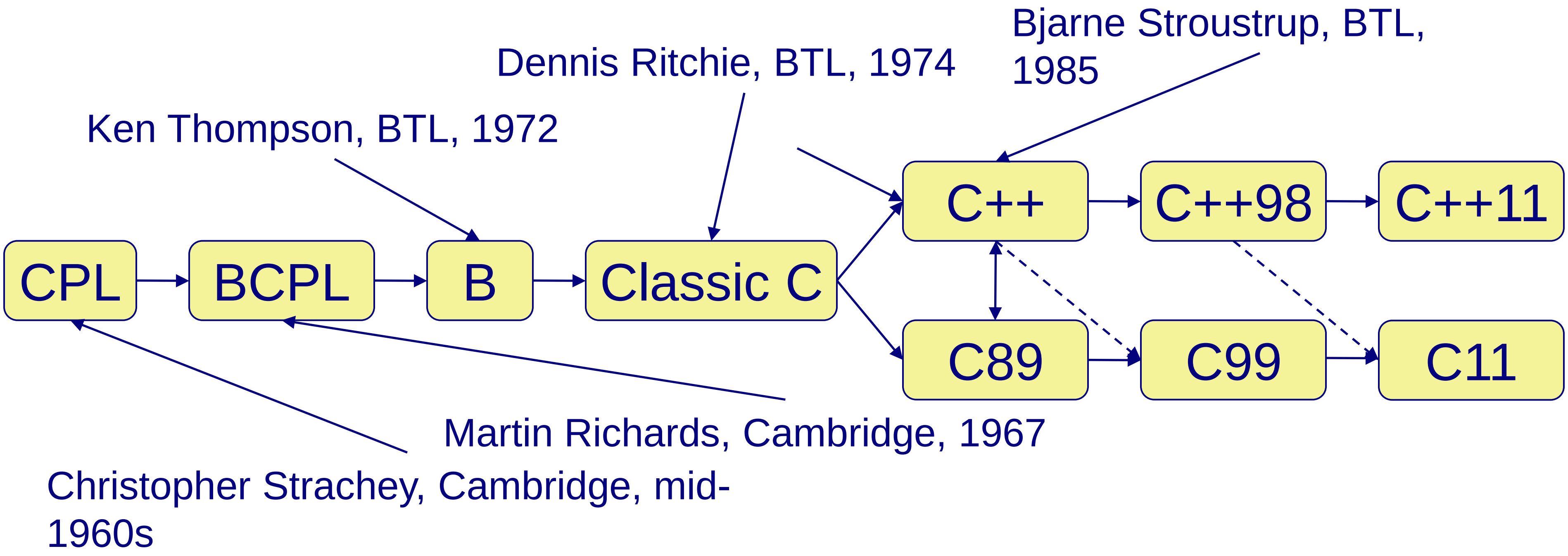
Thompson (left) with Dennis Ritchie.

K. Thompson and D. Ritchie



- Doug McIlroy (b. 1932)
  - Bell Labs
  - Everybody's favorite critic, discussion partner, and ideas man (influenced C, C++, Unix, and much more)

C (1972)


Bjarne Stroustrup, BTL, 1985
Dennis Ritchie, BTL, 1974
Ken Thompson, BTL, 1972
C++
C++98
C++11
CPL
BCPL
B
Classic C
C89
C99
C11
Martin Richards, Cambridge, 1967
Christopher Strachey, Cambridge, mid-1960s

Modula-2 (1978)

- A better, conceptually uniform successor of **Pascal**.
- designed and developed between 1977 and 1985 by **Niklaus Wirth at ETH Zurich**
- to serve as the **sole programming language for the operating system and application software for the personal workstation Lilith**
- Mechanisms to **program concurrency** (many processes running in parallel)
- Not used as much as it deserves
- Its successors, **Modula-3 and Oberon**, are even more conceptually appealing, practically useful—and almost not used at all. (They lost the popularity contest with C++.)

Functional Languages (1978)

Functional Languages

- John Backus (1978) — Turing award lecture

ML

- Edinburgh
- initially designed as meta-language for theorem proving
- Hindley-Milner *type inference*
- "non-pure" functional language (with assignments/side effects)

Miranda, Haskell

- "pure" functional languages with *"lazy evaluation"*

“Hello World” in Functional Languages

SML

```
print("hello world!\n");
```

Haskell

```
hello() = print "Hello World"
```

Ada(1975 – 1985)

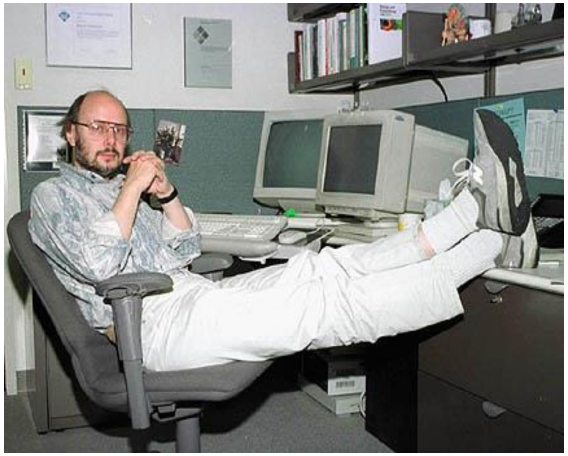
- Designed according to specifications developed by **US Dept of Defense** by a team led by **Jean Ichbiah** of **CII Honeywell Bul**
- Requirements stressed structural programming methodology and readability over writability
- **Renamed Ada (OO) in May 1979** (Reference manual, Mil. Std. 1815 approved 10 December 1980. Ada Bryon was born 10/12/1815)
- Development period 1975 – 1985
  - 1975: first requirements documents
  - 1980: complete language proposed
  - **1983: final standardized version**
  - 1985: working usable compilers appeared
- Contains virtually all elements of PL design
  - Exception handling
  - Parallel processing
  - Abstract data types
  - There are, however, two standards: Ada 83 (the original), and Ada 95.

Smalltalk (1972-80)

- Developed at **Xerox PARC** by **Alan Kay and colleagues (esp. Adele Goldberg)** inspired by **SIMULA 67**
- First compilation in 1972 was written on a bet to come up with "the most powerful language in the world" in "a single page of code".
- It is the purest **object-oriented language** ever designed (till now), **cleaner than Java, much cleaner than C++.**
- In 1980, **SMALLTALK 80**, a uniformly object-oriented programming environment became available as the first commercial release of the Smalltalk language
- Pioneered the graphical user interface everyone now uses
- Industrial use continues to the present day
- Comes complete with a graphical interface and an integrated programming environment.

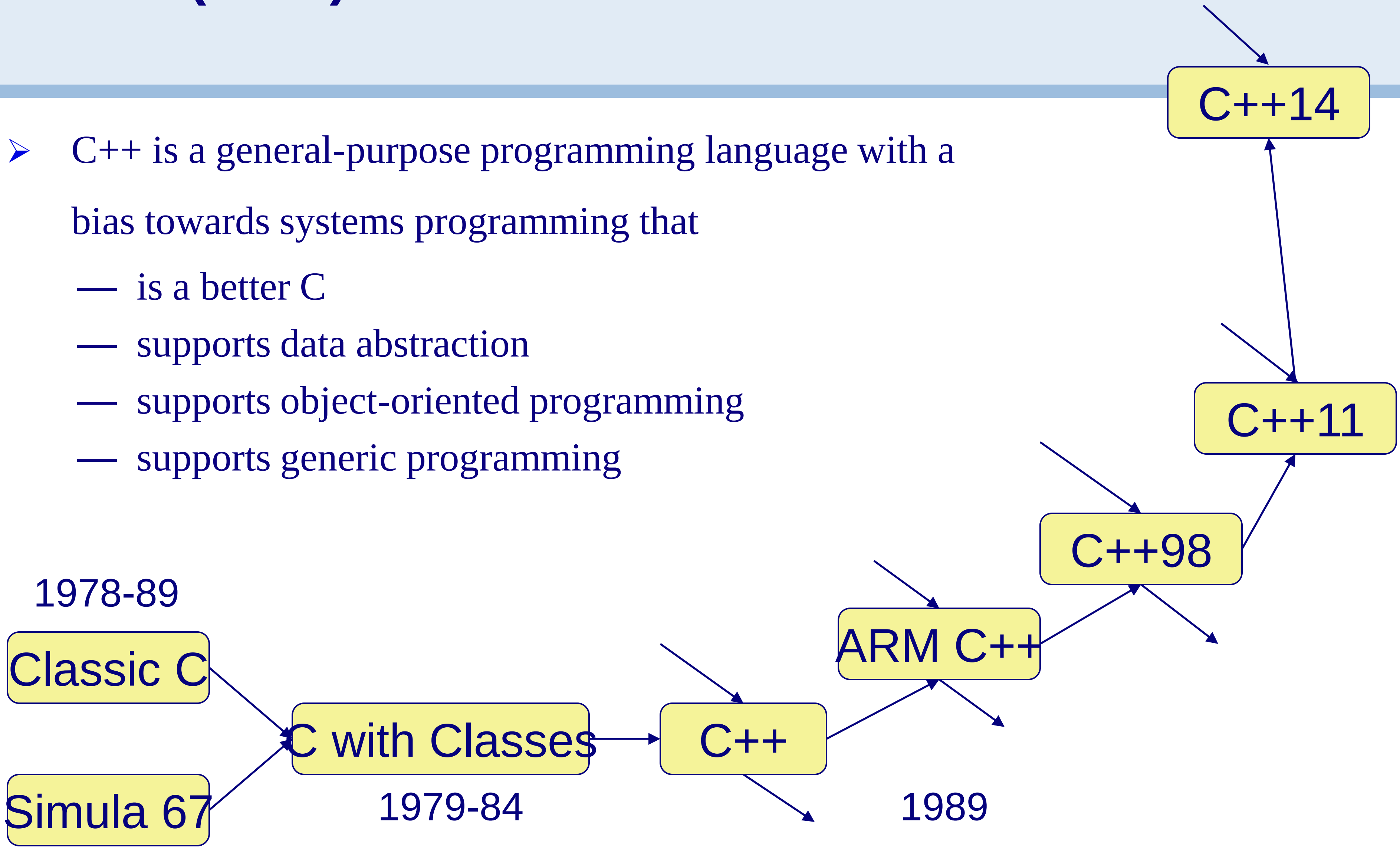
C++ (1986)

- An ***object-oriented*** extension of the *imperative* language C 1985-1986
- Bjarne Stroustrup
  - — AT&T Bell labs
  - — Texas A&M University
- Evolved **from C and SIMULA 67**
- Facilities for object-oriented programming, taken partially from **SIMULA 67, added to C**
- Also has exception handling
- ANSI standard approved in November, 1997
- Complicated syntax, difficult semantics
- Very fashionable, very much in demand
  - — Java did not (yet?) push it out.



C++ (1986)

- C++ is a general-purpose programming language with a bias towards systems programming that
  - — is a better C
  - — supports data abstraction
  - — supports object-oriented programming
  - — supports generic programming



Eiffel (1986)

- Eiffel - a related language that supports OOP

  - Designed by **Bertrand Meyer** (B. 1950-)

  - Not directly derived from any other language

  - Smaller and simpler than C++, but still has most of the power

1990’s: the Internet and Web

- During the **90’s, Object-oriented languages (mostly C++)** became widely used in practical applications
- The Internet and Web drove several phenomena:
  - Adding concurrency and threads to existing languages
  - Increased use of scripting languages such as **Perl, Python and Tcl/Tk, Ruby, JavaScript**

Java (1990-1995)

- Developed at **Sun** in the early 1990s with original goal of a language for embedded computers
- Principals: **Bill Joy, James Gosling, Mike Sheradin, Patrick Naughton**
- Original name, **Oak**, changed for **copyright** reasons
- Based on C++ but significantly simplified
- Supports *only* OOP
- **Has references, but not pointers**
- Includes support for applets and a form of concurrency (i.e. threads)
- Designed for Internet programming, but general-purpose.
- It is said (not quite correctly) to be slow.
- New fashion: maybe the next de-facto standard?

Generations of Programming Languages

1GL: machine codes

2GL: symbolic assemblers

3GL: (machine-independent) imperative languages (FORTRAN, Pascal, LISP, COBOL, ALGOL, Ada, Basic, C, C++, Java,, Smalltalk, …)

4GL: domain specific application (generatorsMaple, Mathematica, Postscript, SPSS, SQL)

5GL: Solve problems using constraints rather than algorithms, used in Artificial Intelligence (Prolog)

How do Programming Languages Differ?

Common Constructs:

> basic data types (numbers, etc.); variables; expressions; statements; keywords; control constructs; procedures; comments; errors ...

Uncommon Constructs:

> type declarations; special types (strings, arrays, matrices, ...); sequential execution; concurrency constructs; packages/modules; objects; general functions; generics; modifiable state; ...

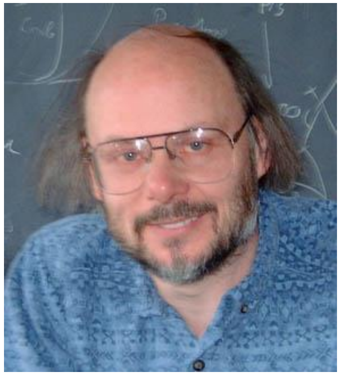
Programming Language Paradigms

- **Procedural**: procedures, sequential execution of code are basic building blocks of program
  - FORTRAN (FORmula TRANslating; John Backus, IBM, 1957)
  - ALGOL (ALGOrithmic Language, 1958)
  - COBOL (COmmon Business Oriented Language, 1960)
  - BASIC (Beginner's All-purpose Symbolic Instruction Code, John Kemeny and Thomas Kurtz, Dartmouth, 1964)
  - Pascal (Niklaus Wirth, 1970)
  - C (Dennis Ritchie, Bell Labs, 1972)

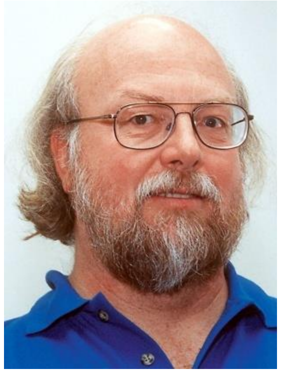
Programming Language Paradigms

- **Object-Oriented**: Program is designed around the ***objects*** required to solve the problem:
  - — Smalltalk (Alan Kay, Xerox PARC, 1971)
  - — Ada (US Dept of Defense, 1975-1993)
  - — C++ (Bjarne Stroustrup, Bell Labs, 1983-1986)
  - — Java (James Gosling, Sun Microsystems, 1990-1995)
  - — C# (Microsoft, 2000)

Alan Kay



B. Stroustrup



J. Gosling

Programming Language Paradigms

- **Functional:** Program is designed around the evaluation of ***functions***, rather than modifying state
  - LISP (John McCarthy, MIT, 1960)
    - *Common Lisp*
    - *Dylan*
    - *Logo*
    - *Scheme*
  - ML (Robin Milner et al, Edinburgh, 1970s)
  - Haskell (purely functional language, 1990)

Programming Language Paradigms

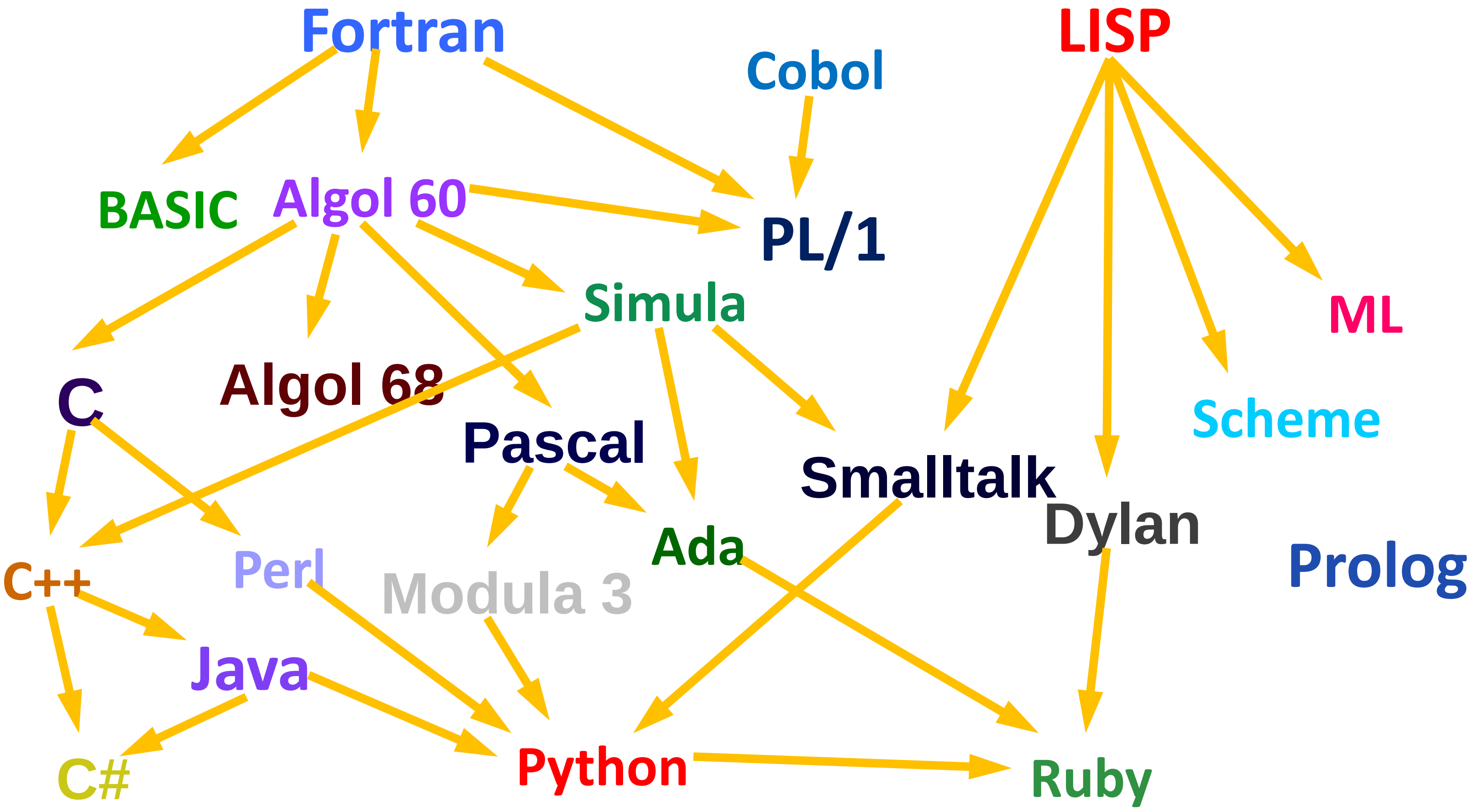
- Logic: Program is declarative, based on *mathematical logic*
  - Prolog (1970)

    A program lists *facts* and *rules*, program execution is controlled deduction to answer a *query*.

Programming Language Paradigms

- Scripting languages (used for text processing, shells, HTML, CGI)
  - awk (Aho, Weinberger, Kerningham, Bell labs, 1978)
  - Perl (Larry Wall, NASA, 1987)
  - Tcl/Tk (John Ousterhout, 1988)
  - Python (Guido van Rossum, CWI, 1991)
  - PHP (Rasmus Lerdorf, 1995)
  - Ruby (Yukihiro Matsumoto, 1996)

A Family Tree Of Languages



A Family Tree of Languages

COBOL
ALGOL
FORTRAN
LISP
APL
PROLOG
PILOT
PL/1
ADA
DOD
PASCAL
MODULA 2
C
UNIX
N.W.
BASIC
FORTH
MACHINE LANGUAGE
ASSEMBLER
LOGO
A⇒B
A B C W
B A 5 8 3 6 7 2