CS213 : Chapter 2 (Evolution of the Major Programming Languages) * اللي بالأخضر مهم*

1. FORTRAN (1954–2008)

- Purpose: Scientific & engineering computations.
- Fortran I (1957):
- Designed for IBM 704; focused on efficiency.
- Features: DO loop, formatted I/O, arithmetic IF, subprograms.
- No dynamic storage or string handling.
- Fortran II (1958): Added independent compilation.
- Fortran IV (1960–62): Explicit types, logical IF, distributed by ANSI standard (1966).
- Fortran 77 (1978): Strings, IF-THEN-ELSE, loops.
- Fortran 90/95/2003/2008: Modules, pointers, dynamic arrays, recursion, OOP, parallel execution, support concurrency.
- Fortan Evaluation: Introduced optimizing compilers; changed scientific programming forever.

2. LISP (1958)

- Purpose: Artificial Intelligence and list processing.
- Features: Functional programming, no variables/assignment.
- Syntax is based on lambda calculus.
- Two data types: uses atoms and lists.
- Control through recursion and Conditional expression.
- Dialects: Common Lisp, Scheme.
- Other functional languages: ML, Haskell, F#.

3. ALGOL Family

- ALGOL 58: Introduced formal types, flexible arrays, begin...end, and :=.
- ALGOL 60:
- Added block structure, recursion, stack-dynamic arrays.
- No I/O (machine dependent).
- Defined syntax with BNF (Backus–Naur Form).
- Foundation for C, Pascal, Java.
- ALGOL 68:

- Added user-defined structures, references, dynamic arrays.
- Less popular, but influenced C, Pascal, Ada.

4. COBOL (1960)

- Purpose: Business and data-processing applications.
- Design: From FLOW-MATIC; English-like syntax.
- Features: Long names, data division (records), nested IFs.
- Contributions:
- First macro facility.
- Hierarchical data structures (Records).
- Required by U.S. Department of Defense (DoD).
- Still used in business systems.

5. BASIC (1964)

- Purpose: Easy for non-science students.
- ~~Designed by: Kemeny & Kurtz (Dartmouth).~~
- Features: Simplicity, friendly interaction, time-sharing support.
- Modern dialect: Visual BASIC.

6. SIMULA 67

- Purpose: Simulation systems.
- Features: Classes, objects, inheritance, and co-routines.
- Contribution: First object-oriented language.

7. Pascal (1971)

- Purpose: Teaching structured programming.
- Features: Simplicity, structured control.
- Used widely in education until 1990s.

8. C (1972)

- Evolved from: B and BCPL, influenced by ALGOL 68.
- Purpose: System programming, UNIX.
- Features: Operators, low-level access, weak type checking.
- Influence: Most later languages (C++, Java, C#).

9. Prolog (1972)

- Type: Logic programming.
- Based on: Formal logic and inference.
- Purpose: AI and reasoning systems.
- Non-procedural: Solves problems using logical rules, not steps.

10. Ada (1980s)

- Features:
- Packages (data abstraction),
- Exception handling,
- Concurrency (tasking model),
- Generic units.
- Ada 95: Added OOP, better concurrency, flexible libraries.
- Decline: DoD dropped requirement; C++ became dominant.

11. Smalltalk (1972–1980)

- First full OOP language (classes, inheritance, dynamic binding)

12. C++ (1980)

- Evolved from: C and SIMULA 67.
- Features: Supports both procedural and OOP styles.
- Complex but powerful.

13. Other OOP Languages

- Objective-C: Based on C + Smalltalk syntax (used by Apple).
- Delphi: Object Pascal; simpler than C++.
- Go (2009): From Google, C-like but not purely OOP.

14. Java (1995)

- Based on: Simplified C++.
- Features:
- OOP only
- Portable
- Strong in Web and app development.

- Impact: Fastest-growing language of its time.

15. Web Scripting Languages

Language	Key Features
Perl (1987)	Text processing, CGI, system admin
JavaScript (1995)	Client-side scripting, dynamic HTML
PHP	Server-side scripting, web forms, DB access
Python	Interpreted, OOP, easy syntax
Ruby	Pure OOP, interpreted
Lua	Lightweight, extensible scripting

16. C# (2000s)

- Based on: C++ and Java.
- Focus: Component-based software and shared type system (CTS).
- Modern enterprise and web development language.

17. Markup/Programming Hybrids

- XSLT: For transforming XML.
- JSP (Java Server Pages): Combines HTML with Java code for dynamic web pages.

Grouped by Decade اللغات من الأقدم إلى الأحداث(التواريخ غير مهمة –مو متأكدة– ولكن كتبته للتوضيح)

1950s–60s

- FORTRAN (1957 → I, II, IV, 77, 90, 95, 2003, 2008)
- LISP (1958)
- ALGOL (58/60/68)
- COBOL (1960)

- BASIC (1964)
- SIMULA 67 (1967 → first OOP)

1970s

- Pascal (1971)
- C (1972)
- Prolog (1972 → logic)
- Smalltalk (1972–1980 → OOP GUI pioneer)

1980s

- Ada (1980s, with later Ada 95/2005)
- C++ (1980)
- Objective-C (early 1980s)
- Delphi (late 1980s–1990s)

1990s

- Java (1995)
- Perl (1987 → popular in 90s)
- JavaScript (mid-90s)
- PHP (mid-90s)
- Python (popularized 1990s)
- Ruby (1990s)
- Lua (1990s)

2000s+

- C# (2000s → .NET era)
- Go (2009)

Grouped by Category أهم نقطة في هذا التشابتر (فئات لغات البرمجة ومثال لكل فئة)

Category	Languages	Notes
Scientific / Systems	FORTRAN, ALGOLs, C, Ada, Go	Performance-oriented, efficient
Business	COBOL	File & record processing
Teaching	BASIC, Pascal	Simplicity for learners
Functional / Logic / AI	LISP, Prolog, Shceme	Symbolic & inference-based
OOP Core	SIMULA, Smalltalk, C++, Objective-C, Delphi, Java, C#, Ruby	Object-oriented design & reuse
Web / Scripting	Perl, Python, PHP, JavaScript, Ruby, Lua	Interpreted & flexible
Markup / Hybrid	XSLT, JSP	Web document transformat