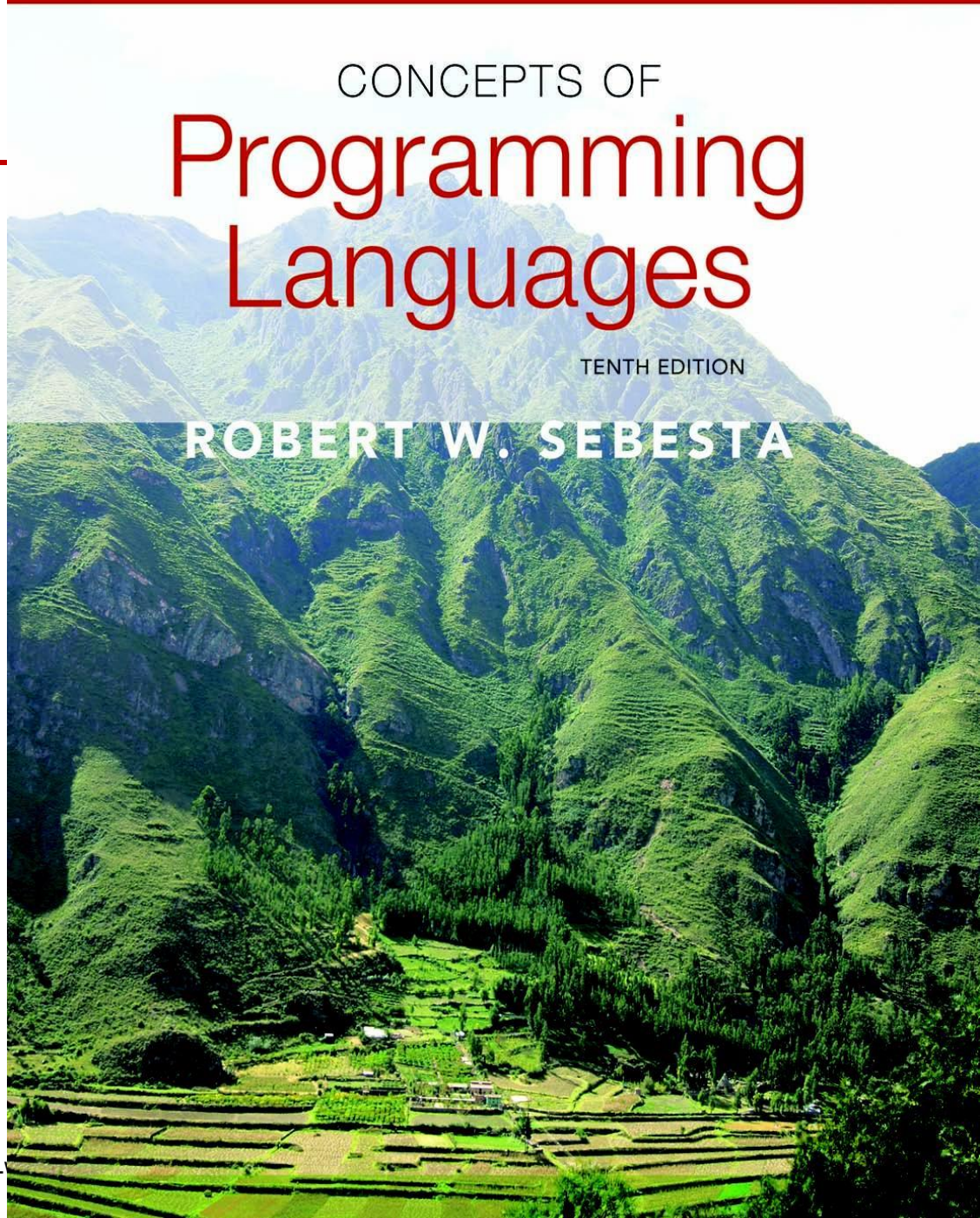
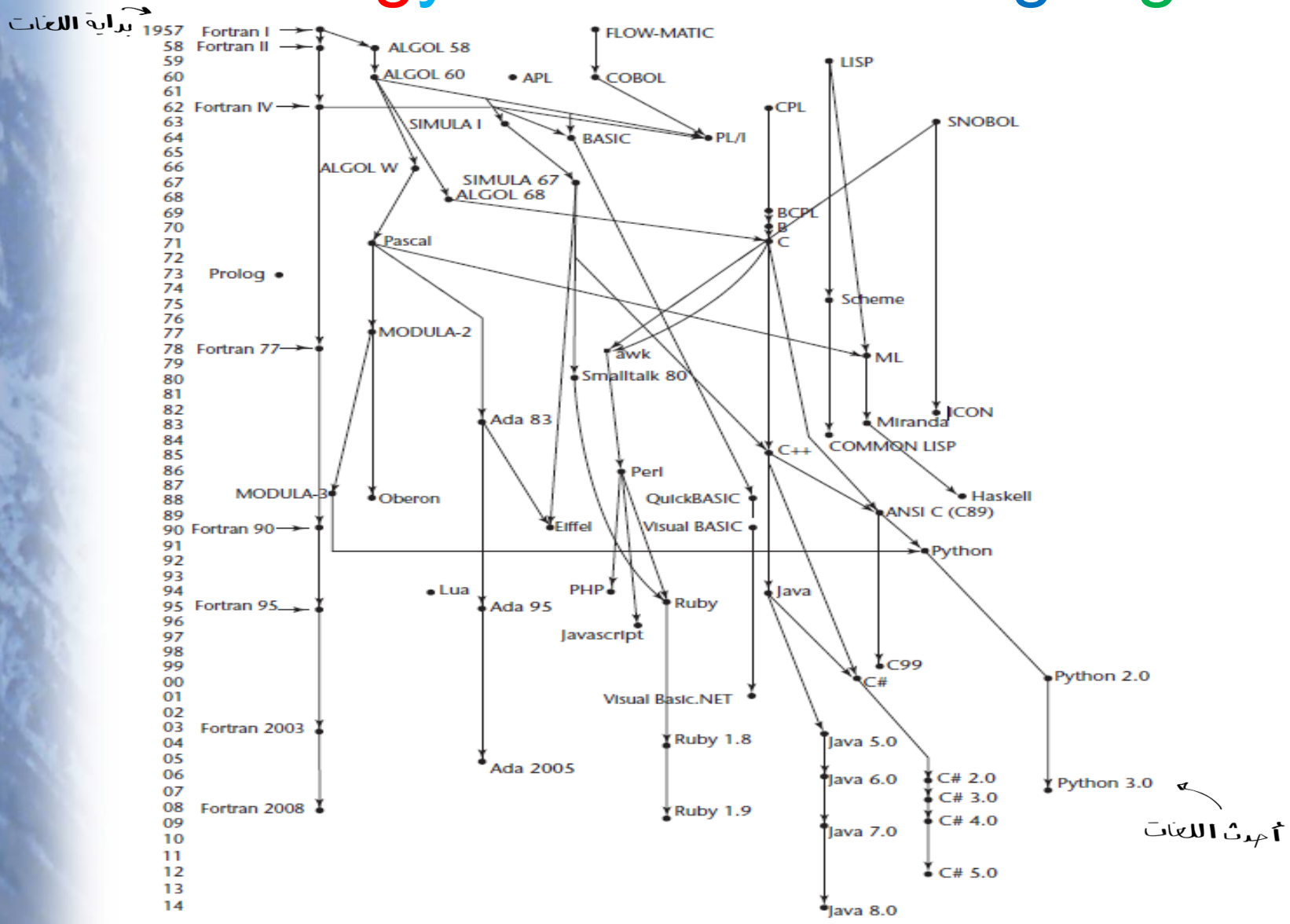
Chapter 2

Evolution of the Major Programming Languages



Copyright © 2021 Pearson Addison-

Genealogy of Common Languages



Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

FORTRAN لغة

لم ينفذ

- Fortran 0: 1954 - not implemented
- Fortran I:1957 ← نقطة البداية

(1) – Designed for the new IBM 704, which had index registers and floating point hardware

(2) – Environment of development:

- Computers were small and unreliable
- Applications were scientific
- No programming methodology or tools
- Machine efficiency was most important

بيئات التطوير : الكمبيوترات كانت صغيرة وغير معتمد عليها، البرامج كانت علمية، ماكان في طرق أو أدوات للبرمجة، كان من المهم أن يكون الجهاز ذو كفاءة

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Design Process of Fortran

- Impact of environment on design of FORTRAN I
  - No need for dynamic storage
  - Need good array handling and counting loops
  - No string handling, decimal arithmetic, or powerful input/output (commercial stuff الاشياء التجارية)

very Primitive

لا يوجد Dynamic storage (تخزين مؤقت ينتهي عند إنتهاء تنفيذ البرنامج)
يوجد Arrays و Loops
لا يوجد تعامل مع نصوص strings أو حسابات عشرية وفواصل

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Fortran I Overview

أول نسخة من الـFortan تم تنفيذه

- First implemented version of FORTRAN

خواص Fortan I :

  - Names could have up to six characters
  - Post-test counting loop (**DO**)
  - Formatted I/O
  - User-defined subprograms
  - No data typing statements
  - Three-way selection statement (arithmetic **IF**)
  - IF (arithmetic expression) N1, N2, N3

```
if (X-1) 10, 20, 30
10 print *, X, " is less than 1"
go to 40 NOT READABLE
20 print *, X, " is equal to 1"
go to 40
30 print *, X, " is more than 1"
```

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Fortran I Overview (continued)

- The first implemented version of FORTRAN
  - No separate compilation
  - The compiler was released in April 1957, after 18 worker-years of effort
  - Programs larger than 400 lines rarely compiled correctly, mainly due to the poor reliability of the IBM 704
  - The code was very fast
  - Quickly became widely used

الكومبايلر أطلق في ١٩٥٧ بعد ١٨ سنة من العمل عليه، طبعًا مع ذلك البرامج اللي تحتوي على أكثر من ٤٠٠ سطر نادر إنه ينفذه كويس بسبب الـIBM704 اللي غير موثوق فيه
الأكواد هنا سريعة بالـFORTAN وهذاك الوقت بسرعة صار إستخدام اللغة شائع

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Fortran II

- Distributed in 1958
  - Independent compilation
  - Fix the bugs

النسخة الثانية من Fortan :
أطلقت عام 1958
كومبايلر مستقل
تصلح الأخطاء

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Fortran IV

- Evolved during 1960-62
  - Explicit type declarations
  - Logical selection statement
  - Subprogram names could be parameters
  - American National Standards Institute (ANSI) standard in 1966

النسخة الثالثة من Fortan :
- طُور في عام ١٩٦٠-١٩٦٢
- data type declarations واضح وصريح
- logical selection زي And Or Not
- يمكن أن تكون أسماء البرامج الفرعية Parameter
اطلقت هذي النسخة بواسطة ANSI عام ١٩٦٦

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Fortran 77

- Became the new standard in 1978
  - Character string handling
  - Logical loop control statement
  - **`IF-THEN-ELSE`** statement

أُطلقت عام ١٩٧٨ Fortan 77
خصائصه:
صار في تعامل مع النصوص والأحرف
- صار في loops
صار في if then else

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Fortran 90

- Most significant changes from Fortran 77
  - Modules
  - Dynamic arrays
  - Pointers
  - Recursion
  - **`CASE`** statement
  - Parameter type checking

نسخة Fortan 90
صار فيها تغييرات عن Fortan 77
وهذا اللي صار فيها : ـــــــــــــــــــ>

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Latest versions of Fortran

النسخ الأخيرة من Fortan :

- Fortran 95 – relatively minor additions, plus some deletions

Objects oriented programming (البرمجة الكائنية)

- Fortran 2003 – support for OOP, procedure pointers, interoperability/compatibility with C
- Fortran 2008 – blocks for local scopes, co-arrays, **Do Concurrent** to specify loops without interdependencies.

تعمل بشكل متوازي منفصل

التزامن : مثلًا أنتي فاتحه الواتس أب والإنستقرام بنفس الوقت

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Fortran Evaluation تقييم لغة الـFortán

حسنت من اداء الكومبايلرز

- Highly optimizing compilers (all versions before 90) محسنه
  - Types and storage of all variables are fixed before run time. تخزين المتغيرات وأنواع المتغيرات ثابت قبل تشغيل البرنامج
- Dramatically changed forever the way computers are used غيرت الى الافضل طريقة استخدام اجهزة الكمبيو

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Functional Programming: LISP

نوع من أنواع البرمجة الإجرائيه أو الوظيفية : LISP

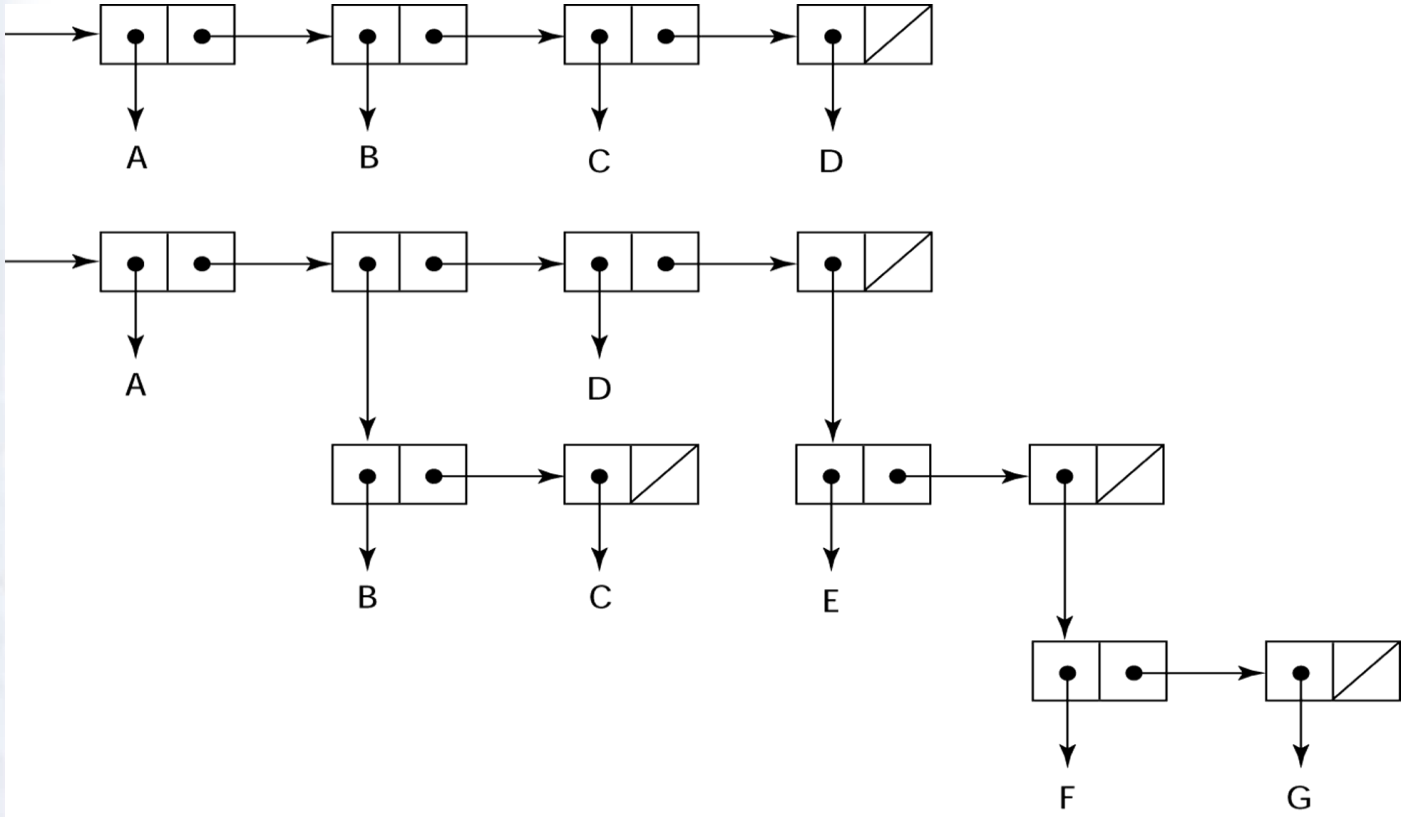
تعتبر أول لغة إجرائية في عالم لغات البرمجة

- The first functional programming language was invented to provide language features for list processing, the need for which grew out of the first applications in the area of artificial intelligence (AI).
- (Designed at MIT by McCarthy)
- AI research needed a language that:
  - Process data in lists (rather than arrays)
  - Symbolic computation (rather than numeric)
- Only two data types: atoms and lists
- Syntax is based on lambda calculus

تستخدم الـList بدال الـArrays
تستخدم الرموز بدال الأرقام

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Representation of Two LISP Lists



Representing the lists `(A B C D)` and `(A (B C) D (E (F G)))`

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

LISP Evaluation تقييم لغة الـLisp

- Pioneered/leading functional programming
  - No need for variables or assignment ما تحتاج إلى متغيرات أو إسناد قيم
  - Control via recursion and conditional expressions
    لتحكم يكون من خلال إستخدام الـRecursion (شيء يعيد إستخدام نفسه) و التعبيرات الشرطية
- Still the dominant language for AI
  اللغة السائدة للذكاء الإصطناعي
- COMMON LISP and Scheme are contemporary dialects of LISP اللهجات المعاصرة
  أمثلة أخرى للغات الـFunctional
- ML, Haskell, and F# are also functional programming languages, but use very different syntax.

[ضروري تعرفون كل فئة من لغات البرمجة وأنواعه]

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

ALGOL 58

- Concept of type was formalized
- Names could be any length أسماء المتغيرات تحتوي على أي طول
- Arrays could have any number of subscripts
- Parameters were separated by mode (in & out)
- Subscripts were placed in brackets, such as

```
myList = [3, 5.8, "grape"]
x = myList[1]      // value of x is ?
```

- Compound statements (**`begin ... end`**) بدال ما نكتب {} لبلوك معين، نكتب begin …end كـ أقواس
- Semicolon as a statement separator زي كل اللغات ننهي الـ Statement بـ ;
- Assignment operator was :=
- **`if`** had an **`else-if`** clause

بالعادة نستخدم علامة يساوي عشان نسند قيمة لمتغير، ولكن في الـalgol نكتبه زي كذا :=

subscripts. In Perl For example,

```
@list[1..5] = @list2[3, 5, 7, 9, 13];
```

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

ALGOL 58 Implementation

- Not meant to be implemented, but variations of it were (MAD, JOVIAL)*
- Although IBM was initially enthusiastic/keen, all support was dropped by mid 1959.

* Jules' Own Version of the International Algebraic Language(JOVIAL).

Michigan Algorithm Decoder(MAD), is a programming language and compiler for the IBM 704

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

ALGOL 60 Overview

- Modified ALGOL 58 at 6-day meeting in Paris
- New features : مميزاته
  - Block structure (local scope)
  - Two parameter passing methods
  - Subprogram recursion
  - Stack-dynamic arrays
  - Still no I/O and no string handling

I/O were omitted because they were thought to be Machine-dependent.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

ALGOL 60 Evaluation

- Successes نجاحاته:
  - It was the standard way to publish algorithms for over 20 years
  - All subsequent imperative languages* are based on it
  - First machine-independent language
  - First language whose syntax was formally defined based on Backus-Naur form(BNF) Chap 3

* Such as C, C++, C#, OOP, and Java, computations are usually specified with variables and assignment statements.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

ALGOL 60 Evaluation (continued)

- Failure
  - Never widely used, especially in U.S.
  - Reasons
    - Lack of I/O and the character set made programs non-portable.
    - Too flexible
    - Hard to implement and inefficient
    - Formal syntax description
    - Lack of support from IBM

فشل هذي اللغة كان بسبب انه ما أستخدمت بشكل واسع،
لأن ماكان في input output, البرنامج غير محمول
يعني ما ينقل من جهاز لآخر، أو نظام تشغيل إلى آخر،
مرن بشكل مبالغ فيه، صعب التنفيذ فيه، مادعمته الـIBM
بشكل كافي

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

COBOL - 1960

- COBOL is a HLPL first developed by the CODASYL Committee (**Co**nference on **Da**ta **Sy**stems **L**anguages) in 1960. Since then, responsibility for developing new COBOL standards has been assumed by the American National Standards Institute (ANSI).
- The word **COBOL** is an acronym that stands for Common Business Oriented Language.
- COBOL was designed for developing business, typically file-oriented, applications. It is not designed for writing systems programs. For instance, you would not develop an operating system or a compiler using COBOL.

انشئت لمجال الأعمال، ما ما نستخدمه عشان نسوي نظام تشغيل

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

COBOL Historical Background

خواص الـ Flow Matic ؟

- Based on FLOW-MATIC*
- FLOW-MATIC features:
  - All user-defined names should be written under constraints.
  - English names for arithmetic operators (no arithmetic expressions)
  - Data and code were completely separate
  - Verbs were first word in every statement

الأسماء المعرفة من قبل المستخدم تكون تحت قيود

العبارات الرياضية + - * / نكتبه حرفًا مثل Add

البيانات والكود نفسه مفصولات عن بعض، زي أيام برمجة ٣ كنا نفصل الملفات، الماين بملف و و و

الأفعال verbs كانت أول ما يكتب في كل statement

* See the programming language chart

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

COBOL Design Process

- First Design Meeting (at the Pentagon) - May 1959
- Design goals:
  - Must look like simple English
  - Must be easy to use, even if that means it will be less powerful.
  - Must broaden/expand the base of computer users.
- Design committee members were all from computer manufacturers and DoD branches
- Design Problems: arithmetic expressions? subscripts? Fights among manufacturers.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

COBOL Evaluation

- Contributions:
  - First macro facility in a high-level language
  - Hierarchical data structures (records)
  - Nested selection statements
  - Long names (up to 30 characters), with hyphens
  - Overall, the data division is the strong part of COBOL's design.

في الـC++ كنا نستخدم Struct
هنا نستخدم records
نفس الفكرة بس المسمى غير

صار في nested if
الأسماء مسموح تكتب بأكثر من ٣٠ حرف
أهم نقطة بهذي اللغة هي وجود records

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

COBOL: DoD Influence

- Comments:
  - First language required by DoD; would have failed without DoD
  - Still the most widely used business applications language.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

The Beginning of Timesharing: BASIC

- Designed by Kemeny & Kurtz at Dartmouth University.
- Design Goals:
  - Easy to learn and use for non-science students
  - Must be "pleasant and friendly"
  - Fast turnaround for homework
  - Free and private access
  - User time is more important than computer time
- Current popular dialect: Visual BASIC
- First widely used language with time sharing

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

SIMULA 67-1967

- Designed primarily for system simulation in Norway by Nygaard and DahL
- Based on ALGOL 60 and SIMULA I
- Primary Contributions
  - –Co-routines - a kind of subprogram
  - –Classes, objects, and inheritance

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

ALGOL 68 - 1968

- From the continued development of ALGOL 60, but it is not a superset of that language
- Design is based on the concept of orthogonality?
- Contributions:
  - User-defined data structures
  - Reference types
  - Dynamic arrays (called flex arrays)

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

ALGOL 68

- Comments:
  - Had even less usage than ALGOL 60
  - Had strong influence on subsequent languages, especially Pascal, C, and Ada

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Important ALGOL Descendants حفيدة

- Pascal - 1971 اللغة اللي راح نتعلمه Pascal
  - Designed by Wirth, who quit the ALGOL 68 committee (didn't like the direction of that work) استخدمت بشكل واسع في مجال التدريس، لأنه سهله وبسيطة
  - Designed for teaching structured programming
  - Small, simple, nothing really new
  - From mid-1970s until the late 1990s, it was the most widely used language for teaching programming in colleges.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Pascal - 1971

سلاید مکرر

- Developed by Wirth (a former member of the ALGOL 68 committee)
- Designed for teaching structured programming
- Small, simple, nothing really new.
- Largest impact was on teaching programming
  - From mid-1970s until the late 1990s, it was the most widely used language for teaching programming

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

C - 1972

- Designed for systems programming (at Bell Labs by Dennis Richie)
- Evolved primarily from BCPL and B, but also ALGOL 68
- Powerful set of operators, but poor type checking
- Initially spread through UNIX
- Though designed as a systems language, it has been used in many application areas.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Programming Based on Logic: Prolog- 1972

- Developed at the University of Aix-Marseille, by Comerauer and Roussel, with some help from Kowalski at the University of Edinburgh
- Based on formal logic.
- Non-procedural.
- Can be summarized as being an intelligent database system that uses an inference process to infer/conclude the truth of given queries.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

History's Largest Design Effort: Ada

- Huge design effort, involving hundreds of people, much money, and about eight years.
- Sequence of requirements (1975-1978)
  - (Strawman, Woodman, Tinman, Ironman, Steelman)
- Named Ada after Augusta Ada Byron, the first programmer.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Ada Evaluation

- Contributions
  - Packages - support for data abstraction
  - Exception handling - elaborate
  - Generic program units
  - Concurrency - through the tasking model
- Comments
  - Competitive design
  - Included all features of software engineering and language design.
  - The first compilers were very difficult; the first really usable compiler came nearly five years after the language design was completed.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Ada 95

- Ada 95 (began in 1988)
  - Support for OOP through type derivation
  - Better control mechanisms for shared data
  - New concurrency features
  - More flexible libraries
- Ada 2005
  - Interfaces and synchronizing interfaces
- Popularity suffered because the DoD no longer requires its use but also because of popularity of C++ تراجع شعبيتها

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Object-Oriented Programming: Smalltalk:1972-1980

- Developed at Xerox PARC, initially by Alan Kay, later by Adele Goldberg
- First full implementation of an object-oriented language (data abstraction تجريد البيانات, inheritance, and dynamic type binding)
- Pioneered the graphical user interface everyone now uses. رائده في الواجهات

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Combining Imperative and OOP: C++

- Developed at Bell Labs in 1980
- Evolved from C and SIMULA 67
- Facilities for object-oriented programming, taken partially from SIMULA 67
- A large and complex language, in part because it supports both procedural and OO programming
- Rapidly grew in popularity, along with OOP
- ANSI standard approved in November 1997
- Microsoft's version*: MC++
  - Properties, delegates, interfaces, no multiple inheritance.\

*Managed Extensions for C++ a Microsoft version of C++ in which native code can be mixed with CLI code.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Related OOP Languages

- Objective-C (at early 1980s)
  - C plus support for OOP based on Smalltalk
  - Uses Smalltalk's method calling syntax
  - Used by Apple for systems programs
- Delphi (Borland)
  - Pascal plus features to support OOP
  - More elegant and safer than C++ أكثر أناقة وأكثر أمانا من
- Go (designed at Google - 2009)
  - Loosely based on C, but also quite different
  - Does not support traditional OOP

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Java (1995)

- Developed at Sun in the early 1990s
- Based on C++
  - Significantly simplified (does not include **`struct, union, enum`**, pointer arithmetic, and half of the assignment of C++)
  - Supports *only* OOP
  - Has references, but not pointers
  - Includes support for applets تطبيقات and a form of concurrency

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Java Evaluation

- Eliminated many unsafe features of C++
- Supports concurrency
- Libraries for applets, GUIs, database access
- Portable: Java Virtual Machine concept, JIT compilers.
- Widely used for Web programming.
- Use increased faster than any previous language
- Most recent version, 7, released in 2011.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Scripting Languages for the Web

- Perl
  - Designed by Larry Wall—first released in 1987
  - Variables are statically typed but implicitly declared
  - Three distinctive namespaces, denoted by the first character of a variable's name
  - Powerful, but somewhat dangerous
  - Gained widespread use for Common Gateway Interface (CGI) programming on the Web
  - Also used for a replacement for UNIX system administration language

مثل:
#a
&a
@a
_ age = 5;

- JavaScript
  - Began at Netscape, but later became a joint venture of Netscape and Sun Microsystems
  - A client-side HTML-embedded scripting language, often used to create dynamic HTML documents
  - Purely interpreted
  - Related to Java only through similar syntax

Administration الرئيس هو المتحكم بكل شيء بدال الموظفين

مثل : chrome safari google

- PHP
  - PHP: Hypertext Preprocessor.
  - A server-side HTML-embedded scripting language, often used for form processing and database access through the Web
  - Purely interpreted

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Scripting Languages for the Web

- Python
  - An OO interpreted scripting language = programming language
  - Used for Common Gateway Interface (CGI) programming and form processing
  - Supports lists, tuples, and hashes
- Ruby
  - Designed in Japan by Yukihiro Matsumoto (a.k.a, "Matz")
  - Began as a replacement for Perl and Python
  - A pure object-oriented scripting language
  - All data are objects
  - Most operators are implemented as methods, which can be redefined by user code
  - Purely interpreted.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Scripting Languages for the Web

- Lua
  - An OO interpreted scripting language
  - Used for CGI programming and form processing
  - Supports lists, tuples, and hashes, all with its single data structure, the table
  - Easily extendable

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

C#

- Part of the .NET development platform
- Based on C++ and Java
- Provides a language for component-based software development.
- All .NET languages (C#, Visual BASIC.NET, Managed C++, J#.NET, and Jscript.NET) use Common Type System (CTS), which provides a common class library.
- Likely to become widely used

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Markup/Programming Hybrid Languages

- XSLT
  - eXtensible Markup Language (XML): a metamarkup language
  - eXtensible Stylesheet Language Transformation (XSTL) transforms XML documents for display
  - Programming constructs (e.g., looping)

- JSP
  - Java Server Pages: a collection of technologies to support dynamic Web documents.
  - JSTL, a JSP library, includes programming constructs in the form of HTML elements

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.

Summary

- Development, development environment, and evaluation of a number of important programming languages
- Perspective into current issues in language design.

Copyright © 2021 Pearson Addison-Wesley. All rights reserved.