Languages and Compilers
(SProg og Oversættere)

Lecture 2
Programming Language Evolution

Bent Thomsen

Department of Computer Science

Aalborg University

Learning goals

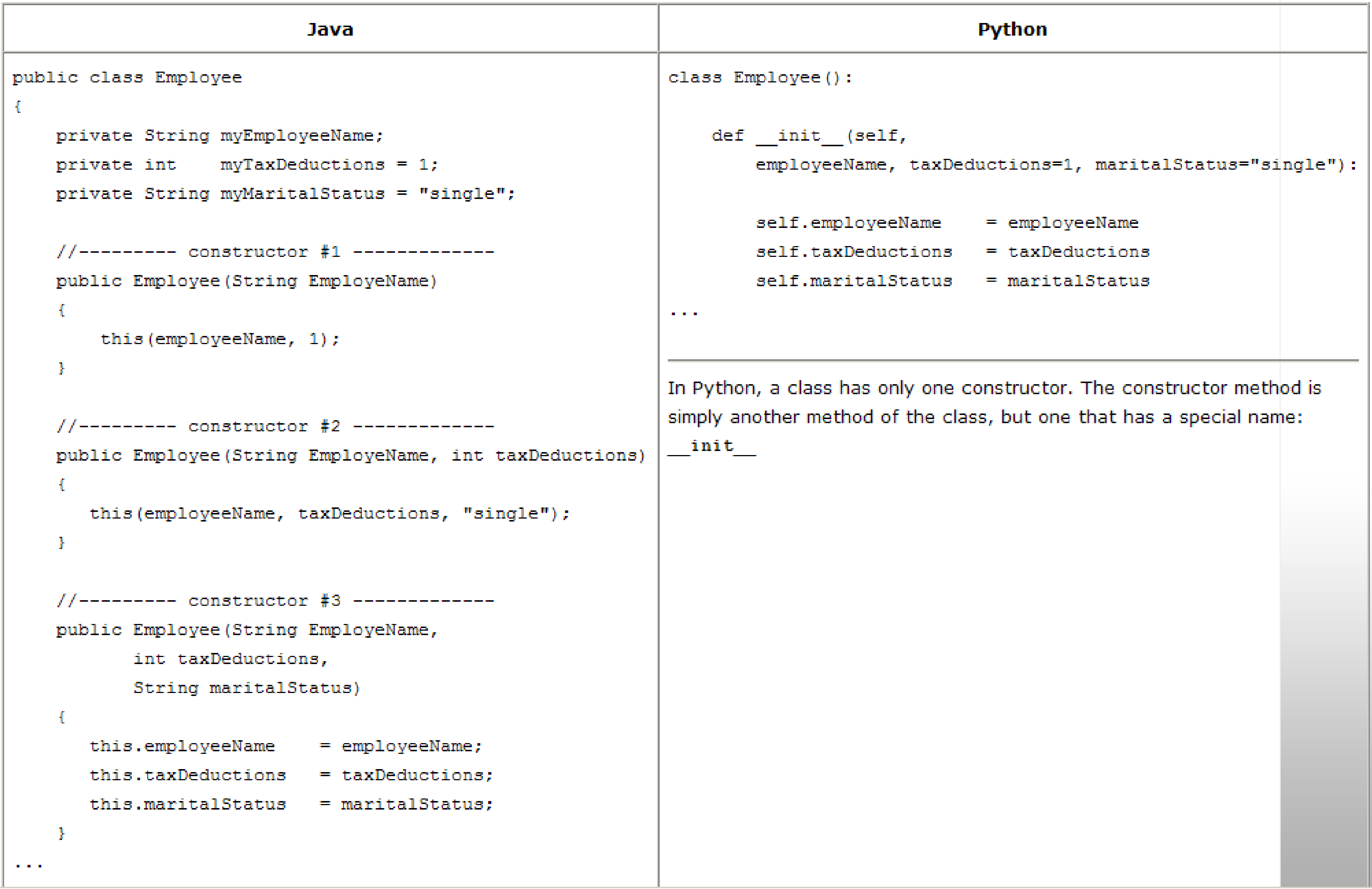
- Introduction to programming language design
- Overview of the evolution of programming languages

Why Are There So Many Programming Languages

- Why does some people speak French?
- Programming languages have evolved over time as better ways have been developed to design them.
  - First programming languages were developed in the 1950s
  - Since then thousands of languages have been developed
- Different programming languages are designed for different types of programs.

Why do people design new programming Languages?

- Most new languages are invented out of frustration!
  - “The decision to create a new programming language or to design an extension of an existing language is often a reaction to some language that the designer knows (and likes or dislikes)”
    - P. Sestoft 2012
- A few languages are created because somebody requested a new language
  - Fortran, C#, Swift, DART
  - All of you, because the study regulations says so ☺



<table>
<tr><th>Java</th><th>Python</th></tr>
<tr><td>

```java
public class Employee
{
    private String myEmployeeName;
    private int    myTaxDeductions = 1;
    private String myMaritalStatus = "single";

    //--------- constructor #1 -------------
    public Employee(String EmployeName)
    {
        this(employeeName, 1);
    }

    //--------- constructor #2 -------------
    public Employee(String EmployeName, int taxDeductions)
    {
       this(employeeName, taxDeductions, "single");
    }

    //--------- constructor #3 -------------
    public Employee(String EmployeName,
           int taxDeductions,
           String maritalStatus)
    {
       this.employeeName    = employeeName;
       this.taxDeductions   = taxDeductions;
       this.maritalStatus   = maritalStatus;
    }
...
```

</td><td>

```python
class Employee():

    def __init__(self,
        employeeName, taxDeductions=1, maritalStatus="single"):

        self.employeeName    = employeeName
        self.taxDeductions   = taxDeductions
        self.maritalStatus   = maritalStatus
...
```

In Python, a class has only one constructor. The constructor method is simply another method of the class, but one that has a special name: **__init__**

</td></tr>
</table>

Programming Language design

- Designing a new programming language or extending an existing programming language usually follows an iterative approach:

1. Create ideas for the programming language or extensions
2. Describe/define the programming language or extensions
3. Implement the programming language or extensions
4. Evaluate the programming language or extensions
5. If not satisfied, goto 1

Programming Language design

1. Create ideas for the programming language or extensions
   - This subject is almost completely absent from literature!
2. Describe/define the programming language or extensions
   - We will spend quite a bit of time in this course and the SS
3. Implement the programming language or extensions
   - We will spend a lot of time on this subject.
4. Evaluate the programming language or extensions
   - is not usually covered in classic litterature on Programming Languages and Compilers!
   - But you saw Sebesta's Language evaluation criteria in the last lecture
   - We shall see a some more later.

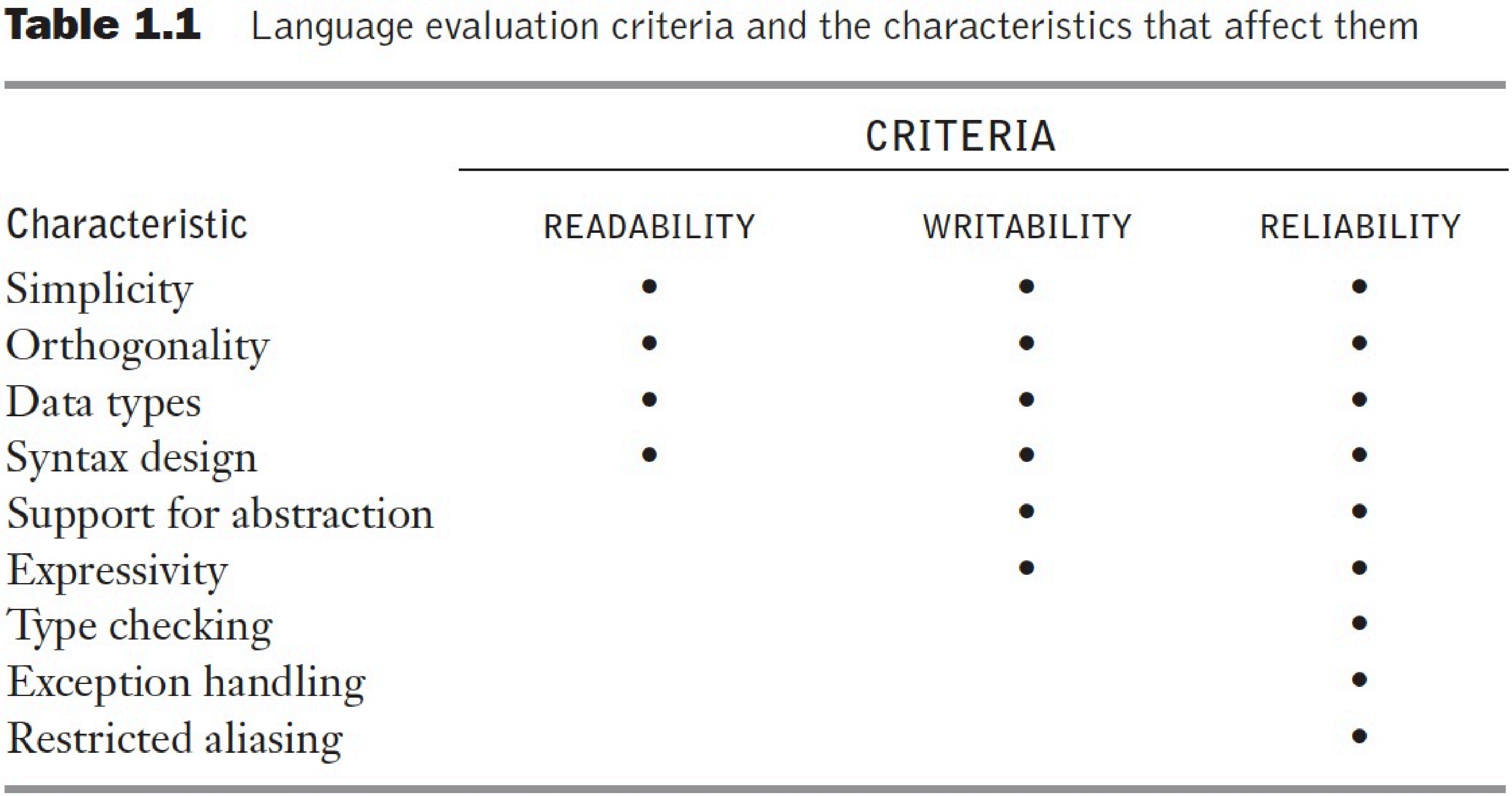


Table 1.1 Language evaluation criteria and the characteristics that affect them

	CRITERIA		
Characteristic	READABILITY	WRITABILITY	RELIABILITY
Simplicity	•	•	•
Orthogonality	•	•	•
Data types	•	•	•
Syntax design	•	•	•
Support for abstraction		•	•
Expressivity		•	•
Type checking			•
Exception handling			•
Restricted aliasing			•

How to create ideas for a new programming language or extensions ?

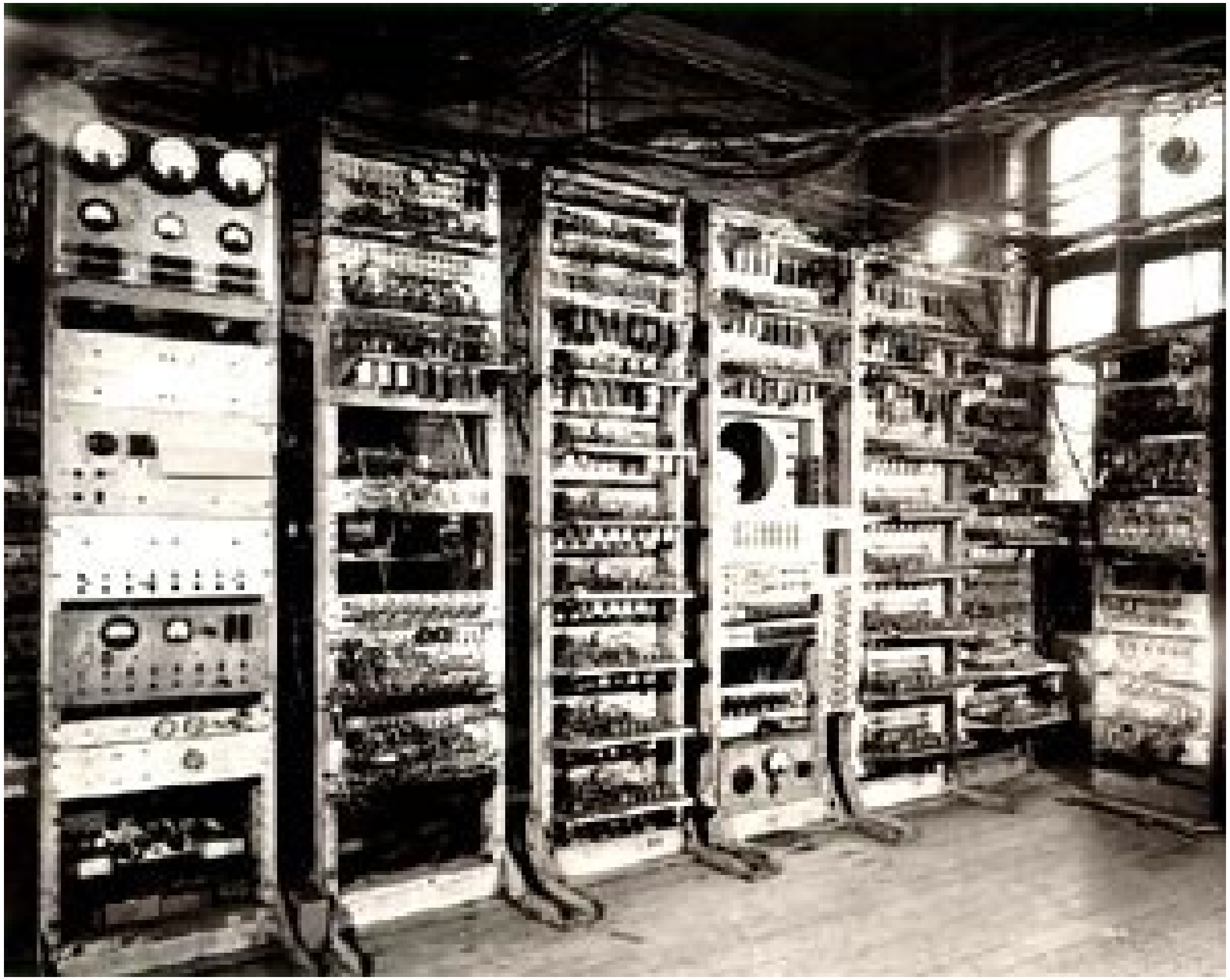
- Do a problem analysis!
  - Who needs the new language?
  - What is the purpose of the new language
  - What type of programs would we like to write?
    - Create some example programs
    - Even before you have defined the language you can create examples of programs as you would like them to look
- Take inspiration from other languages
  - Which langauges do you know?
  - What do you like about these languages?
  - What do you dislike?
  - Look at languages you don’t know!
  - Look at the history of programming languages

Programming Language History
1940s

The first electronic computers were monstrous contraptions

- Programmed in binary *machine code* by hand
- Code is not reusable or *relocatable*
  - *Each machine had its own machine language*
- Computation and machine maintenance were difficult:
  - cathode tubes regularly burned out
  - The term ''*bug*'' originated from a bug that reportedly roamed around in a machine causing short circuits

… in the beginning of time



Programming Language History
Late 1940s early 1950s

- *Assembly languages*
  - invented to allow machine operations to be expressed in mnemonic abbreviations
  - Enables larger, reusable, and re-locatable programs
  - Actual machine code is produced by an *assembler*
  - Early assemblers had a one-to-one correspondence between assembly and machine instructions
  - Later: expansion of *macros* into multiple machine instructions to achieve a form of higher-level programming

Assembly
LOAD x
ADD R1 R2

```
; Hello World for Intel Assembler (MSDOS)

mov ax,cs
mov ds,ax
mov ah,9
mov dx, offset Hello
int 21h
xor ax,ax
int 21h
```

Programming Language History
Mid 1950s

- Fortran , the first higher-level language
  - Now programs could be developed that were machine independent!
  - Main computing activity in the 50s: solve numerical problems in science and engineering
  - Other high-level languages soon followed:
    - Algol 58 is an improvement compared to Fortran
    - Cobol for business computing
    - Lisp for symbolic computing and artificial intelligence
    - BASIC for "beginners"

```
C     Hello World in Fortran

      PROGRAM HELLO
      WRITE (*,100)
      STOP
  100 FORMAT (' Hello World! ' /)
      END
```

```
 * Hello World in COBOL

****************************
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
ENVIRONMENT DIVISION.
DATA DIVISION.
PROCEDURE DIVISION.
MAIN SECTION.
DISPLAY "Hello World!"
STOP RUN.
***************************
```

Programming Language History
1960s

- Structured Programming
  - Dijkstra, Dahl, and Hoare.
- Pascal, Niklaus Wirth (ETH, Zurich)
  - Modelled after Algol
  - No GOTO
  - Very strongly typed
  - Procedures nested inside each other
  - Designed for teaching programming
- Simula, Dahl and Nygaard (Norway)
  - The first language with objects, classes, and subclasses

```
{Hello world in Pascal}

program HelloWorld(output);
begin
  WriteLn('Hello World!');
end.
```

Programming Language History
1970s

- C, Dennis Ritchie/Ken Thompson (Bell Labs)
  - Successor to B, which was stripped-down BCPL.
  - High-level constructs and low-level power
  - Flat name space for functions/procedures
- Ada, Jean Ichbiah (France)
  - Instigated by the Department of Defense
  - Designed for systems programming, especially embedded systems.

```c
/* Hello World in C, Ansi-style */

#include <stdio.h>
#include <stdlib.h>

int main(void)
{
  puts("Hello World!");
  return EXIT_SUCCESS;
}
```

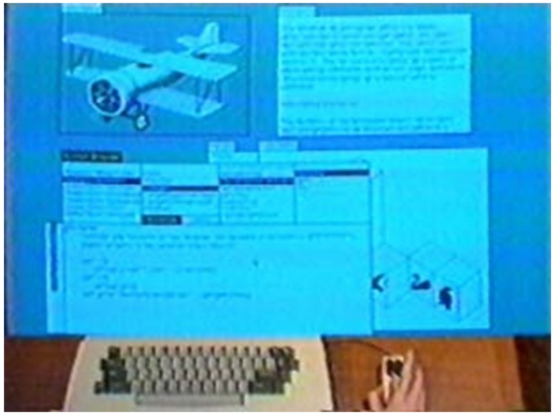
```ada
-- Hello World in Ada

with Text_IO;
procedure Hello_World is

begin
  Text_IO.Put_Line("Hello
World!");
end Hello_World;
```

Programming Language History
1970s

- Smalltalk, Alan Kay, Adele Goldberg (Xerox PARC)
  - Graphics-rich
    - GUI
    - Fonts
  - Object-oriented
    - Everything is an object
    - Objects communicate through messages



- Scheme, Gerald Sussman & Guy Steele (MIT)
  - LISP with static scoping
- Prolog, Philippe Roussel (France)
  - Based on rules, facts, and queries.

```
"Hello World in Smalltalk"

Transcript show: 'Hello
World!'.
```

```
; Hello World in Scheme

(display "Hello, world!")
(newline)
```

```
% Hello World in Prolog

hello :- display('Hello
World!') , nl .
```

Programming Language History 1980s

- Object-oriented programming
  - Important innovation for software development
  - The concept of a class is based on the notion of data type abstraction from Simula 67 , a language for discrete event simulation that has classes but no inheritance
- 1979-1983: C++ Bjarne Stroustrop (Bell Labs)
  - Originally thought of as “C with classes”.
  - First widely-accepted object-oriented language.
  - First implemented as a pre-processor for the C compiler.

```
// Hello World in C++ (pre-ISO)

#include <iostream.h>

main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

Programming Language History
1980s

- Functional Programming
  - Extensive list of new concepts
    - Lazy vs. eager evaluation
    - Pure vs. imperative features
    - Parametric polymorphism
    - Type inference
    - (Garbage collection)
  - Hope
  - Clean
  - Haskell
  - SML
  - Caml

Programming Language History 1990s

- HTML, Tim Berners-Lee (CERN)
  - “Hypertext Markup Language”
    - Language of the World Wide Web.
  - A markup language, not a programming language.
- Scripting languages
  - PERL.
    - CGI or Apache module
  - Languages within Web pages
    - JavaScript, VBScript
    - PHP, ASP, JSP
- Java, James Gosling (Sun)

The evolution of Java

- 1993 Oak project at Sun
  - small, robust, architecture independent, Object-Oriented, language to control interactive TV.
  - didn't go anywhere
- 1995 Oak becomes Java
  - Focus on the web
- 1996 Java 1.0 available
- 1997 (March) Java 1.1 - some language changes, much larger library, new event handling model
- 1997 (September) Java 1.2 beta – huge increase in libraries including Swing, new collection classes, J2EE
- 1998 (October) Java 1.2 final (Java2!)

- 2000 (April) Java 1.3 final
- 2001 Java 1.4 final (assert)
- 2004 Java 1.5 (parameterized types, enum, …)
- 2005 J2EE 1.5
- 2006 Java 6
- 2011 Java 7
- 2014 Java 8 (lambda expressions)
- 2017 Java 9 (expected 23.3.17, but released 21.9.17
- – REPL, process control, collections, streams, …)
- 2018 Java 10 (March – Minor updates, GC interface, parallel GC)
- 2018 Java 11 (September - Local-variable syntax for lambda parameters, ZGC: a scalable low-latency GC)
- 2019 Java 12 (March)
- Java SE 13 (September 17, 2019)
- Java SE 14 (March 17, 2020) – preview of patternmatching
- Java SE 15 (September 15, 2020)

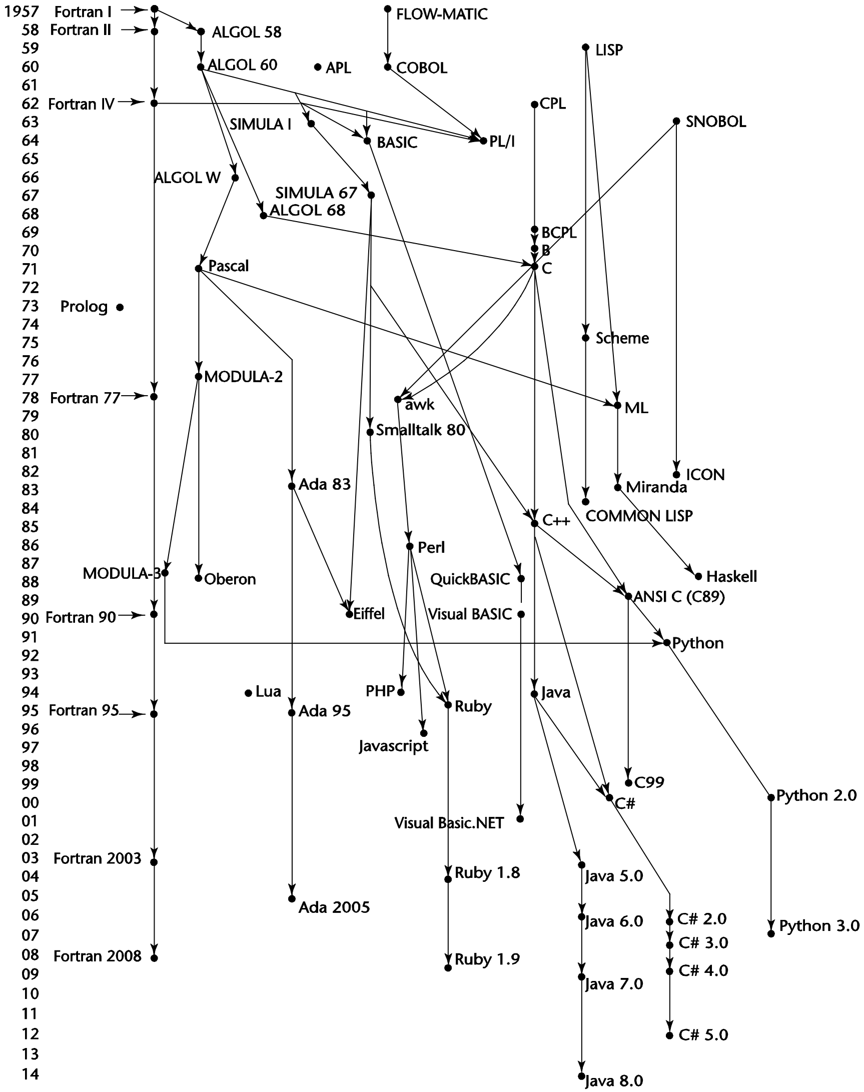
Programming Language History
2000s

- XML
- Microsoft .NET
  - Multiple languages
    - C++
    - C#
    - Visual Basic
    - COBOL
    - Fortran
    - Eiffel
  - Common virtual machine (.Net CLR)
  - Web services

C# History

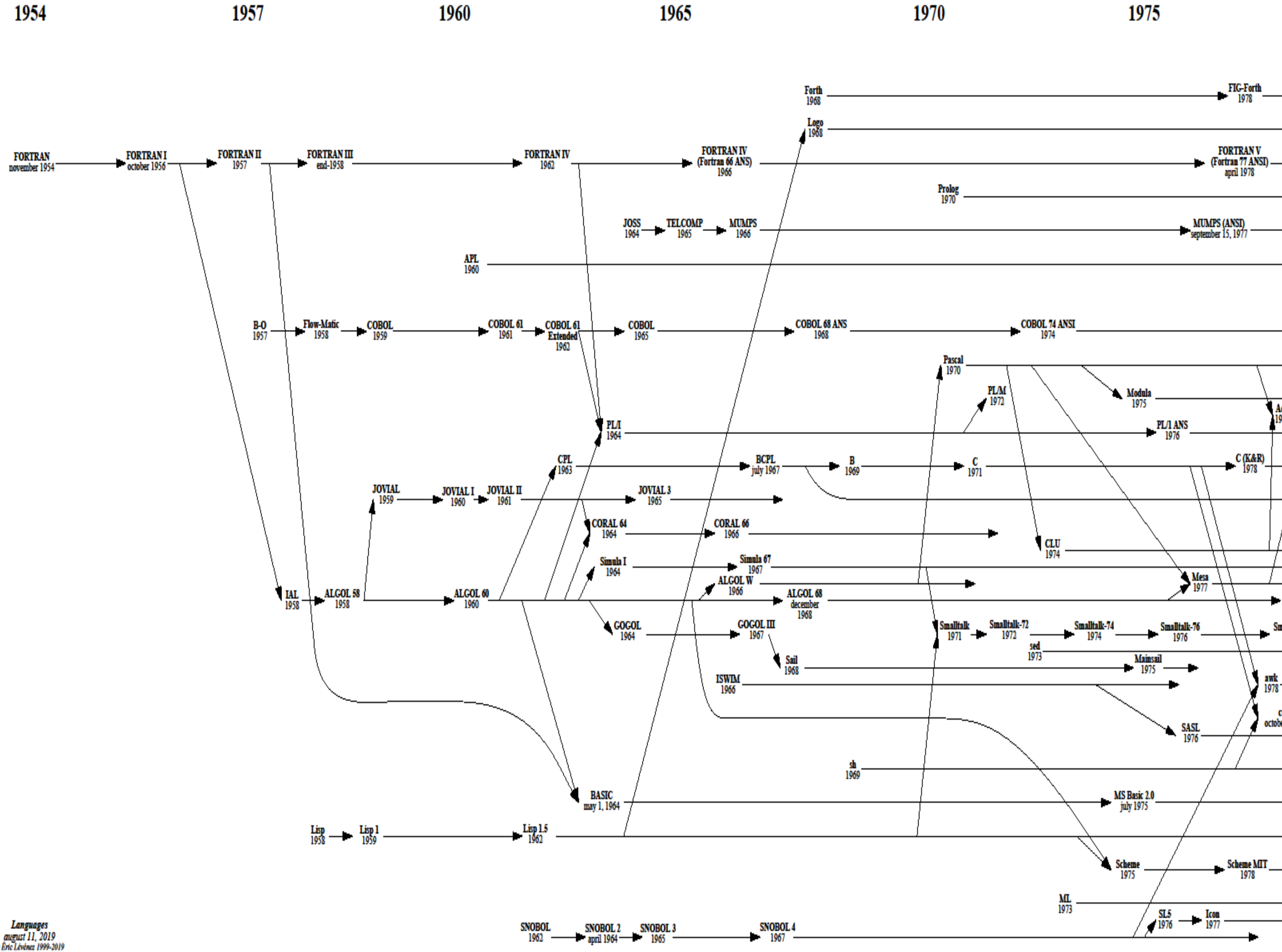
- 12/1998 – COOL project started
- 07/1999 – First internal ports to COOL
- 02/2000 – Named changed to C#
- 07/2000 – First public preview release
- 02/2002 – C# 1.0, VS.NET 2002
- 05/2003 – C# 1.1, VS.NET 2003
- 06/2004 – Beta 1 of C# 2.0 and VS 2005
- 04/2005 – Beta 2 of C# 2.0 and VS 2005
- 11/2005 – C# 2.0 VS 2005, C# 2.0 release
  - Generics, anonymous delegates, nullable types, iterators, partial classes
- 11/2006 – C# 3.0, VS 2008
  - (local type inference, lambdas, expression trees, LINQ)
- 04/2010 – C# 4.0, VS 2010
  - Type dynamics, named+optional parameters, co-/contra variant generics
- 08/2012 – C# 5.0, VS 2012
  - Async methods
- 06/2015 – C# 6.0, VS 2015
  - Await in catch/finally blocks, succinct null checking
- 2017 – C# 7.0,7.1,7.2, VS 2017
  - Pattern matching, Local functions, tuples
- 2018 – C# 7.3
  - Reassigning ref local variables, Using initializers on stackalloc arrays
- 2019 – C# 8
  - readonly struct members, default interface members, switch expressions, Property, Tuple, and positional patterns, using declarations
  - static local functions, Disposable ref struct, Nullable reference types, Indices and Ranges, Null-coalescing assignment,Async Streams
- 2020 – C# 9

Genealogy of Common Languages

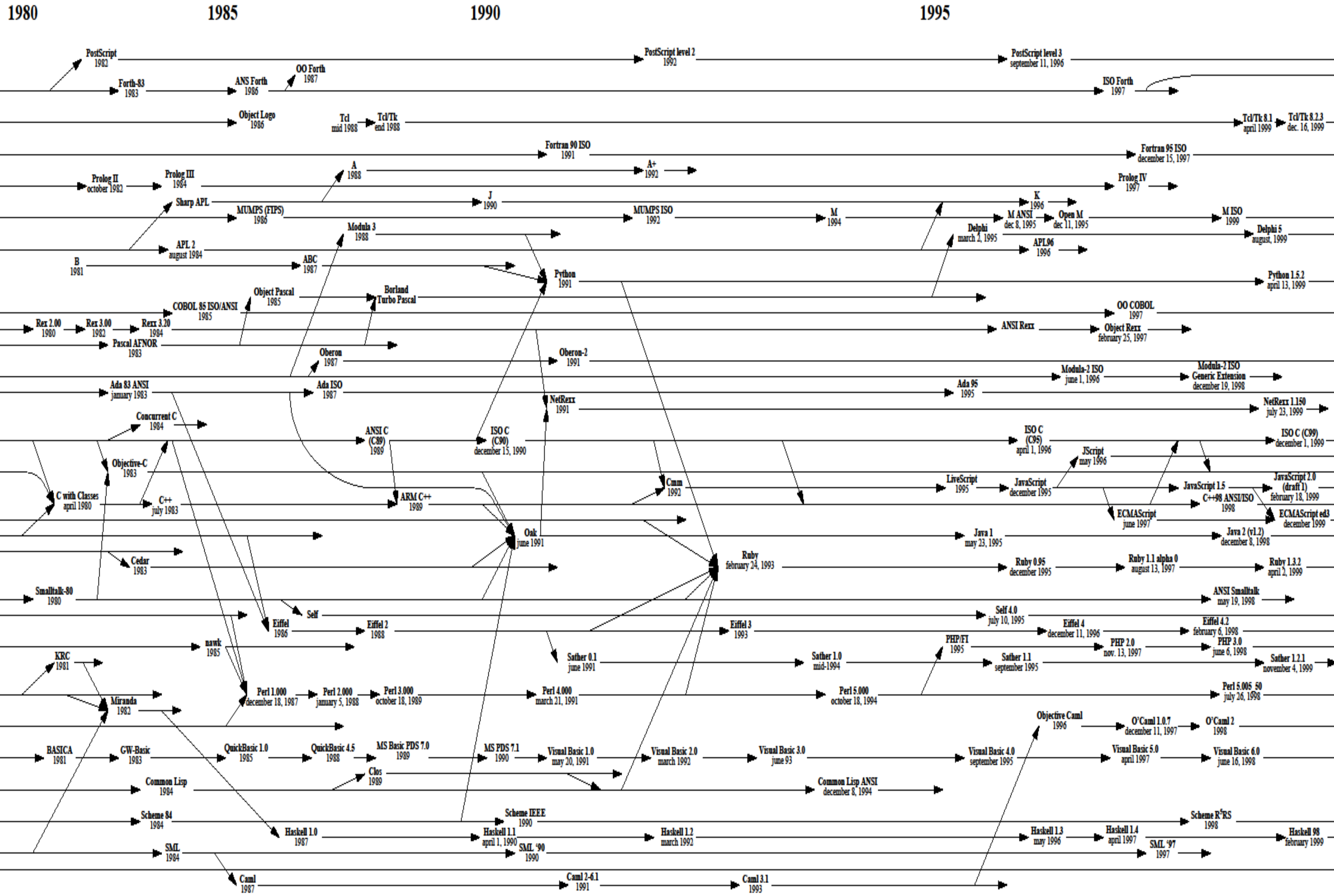


lang.pdf

Copyright © 2017 Pearson Education, Ltd. All rights reserved.



Languages
august 11, 2019
© Éric Lévénez 1999-2019
<http://www.levenez.com/lang/>



1980 1985 1990 1995

PostScript 1982 → PostScript level 2 1992 → PostScript level 3 september 11, 1996

Forth-83 1983 → ANS Forth 1986 → OO Forth 1987; ISO Forth 1997

Object Logo 1986; Tcl mid 1988 → Tcl/Tk end 1988 → Tcl/Tk 8.1 april 1999 → Tcl/Tk 8.2.3 dec. 16, 1999

Fortran 90 ISO 1991 → Fortran 95 ISO december 15, 1997

A 1988 → A+ 1992

Prolog II october 1982 → Prolog III 1984 → Prolog IV 1997

Sharp APL → J 1990 → K 1996

MUMPS (FIPS) 1986 → MUMPS ISO 1992 → M 1994 → M ANSI dec 8, 1995 → Open M dec 11, 1995 → M ISO 1999

Modula 3 1988

Delphi march 2, 1995 → Delphi 5 august, 1999

APL 2 august 1984 → APL96 1996

B 1981 → ABC 1987 → Python 1991 → Python 1.5.2 april 13, 1999

Object Pascal 1985 → Borland Turbo Pascal

COBOL 85 ISO/ANSI 1985 → OO COBOL 1997

Rex 2.00 1980 → Rex 3.00 1982 → Rexx 3.20 1984 → ANSI Rexx → Object Rexx february 25, 1997

Pascal AFNOR 1983

Oberon 1987 → Oberon-2 1991

Modula-2 ISO june 1, 1996 → Modula-2 ISO Generic Extension december 19, 1998

Ada 83 ANSI january 1983 → Ada ISO 1987 → Ada 95 1995

NetRexx 1991 → NetRexx 1.150 july 23, 1999

Concurrent C 1984

ANSI C (C89) 1989 → ISO C (C90) december 15, 1990 → ISO C (C95) april 1, 1996 → ISO C (C99) december 1, 1999

Objective-C 1983

JScript may 1996

C with Classes april 1980 → C++ july 1983 → ARM C++ 1989 → C++98 ANSI/ISO 1998

Cmm 1992

LiveScript 1995 → JavaScript december 1995 → JavaScript 1.5 → JavaScript 2.0 (draft 1) february 18, 1999

ECMAScript june 1997 → ECMAScript ed3 december 1999

Oak june 1991 → Java 1 may 23, 1995 → Java 2 (v1.2) december 8, 1998

Cedar 1983

Ruby february 24, 1993 → Ruby 0.95 december 1995 → Ruby 1.1 alpha 0 august 13, 1997 → Ruby 1.3.2 april 2, 1999

Smalltalk-80 1980 → ANSI Smalltalk may 19, 1998

Self → Self 4.0 july 10, 1995

Eiffel 1986 → Eiffel 2 1988 → Eiffel 3 1993 → Eiffel 4 december 11, 1996 → Eiffel 4.2 february 6, 1998

nawk 1985

PHP/FI 1995 → PHP 2.0 nov. 13, 1997 → PHP 3.0 june 6, 1998

KRC 1981

Sather 0.1 june 1991 → Sather 1.0 mid-1994 → Sather 1.1 september 1995 → Sather 1.2.1 november 4, 1999

Miranda 1982

Perl 1.000 december 18, 1987 → Perl 2.000 january 5, 1988 → Perl 3.000 october 18, 1989 → Perl 4.000 march 21, 1991 → Perl 5.000 october 18, 1994 → Perl 5.005_50 july 26, 1998

Objective Caml 1996 → O'Caml 1.0.7 december 11, 1997 → O'Caml 2 1998

BASICA 1981 → GW-Basic 1983 → QuickBasic 1.0 1985 → QuickBasic 4.5 1988 → MS Basic PDS 7.0 1989 → MS PDS 7.1 1990 → Visual Basic 1.0 may 20, 1991 → Visual Basic 2.0 march 1992 → Visual Basic 3.0 june 93 → Visual Basic 4.0 september 1995 → Visual Basic 5.0 april 1997 → Visual Basic 6.0 june 16, 1998

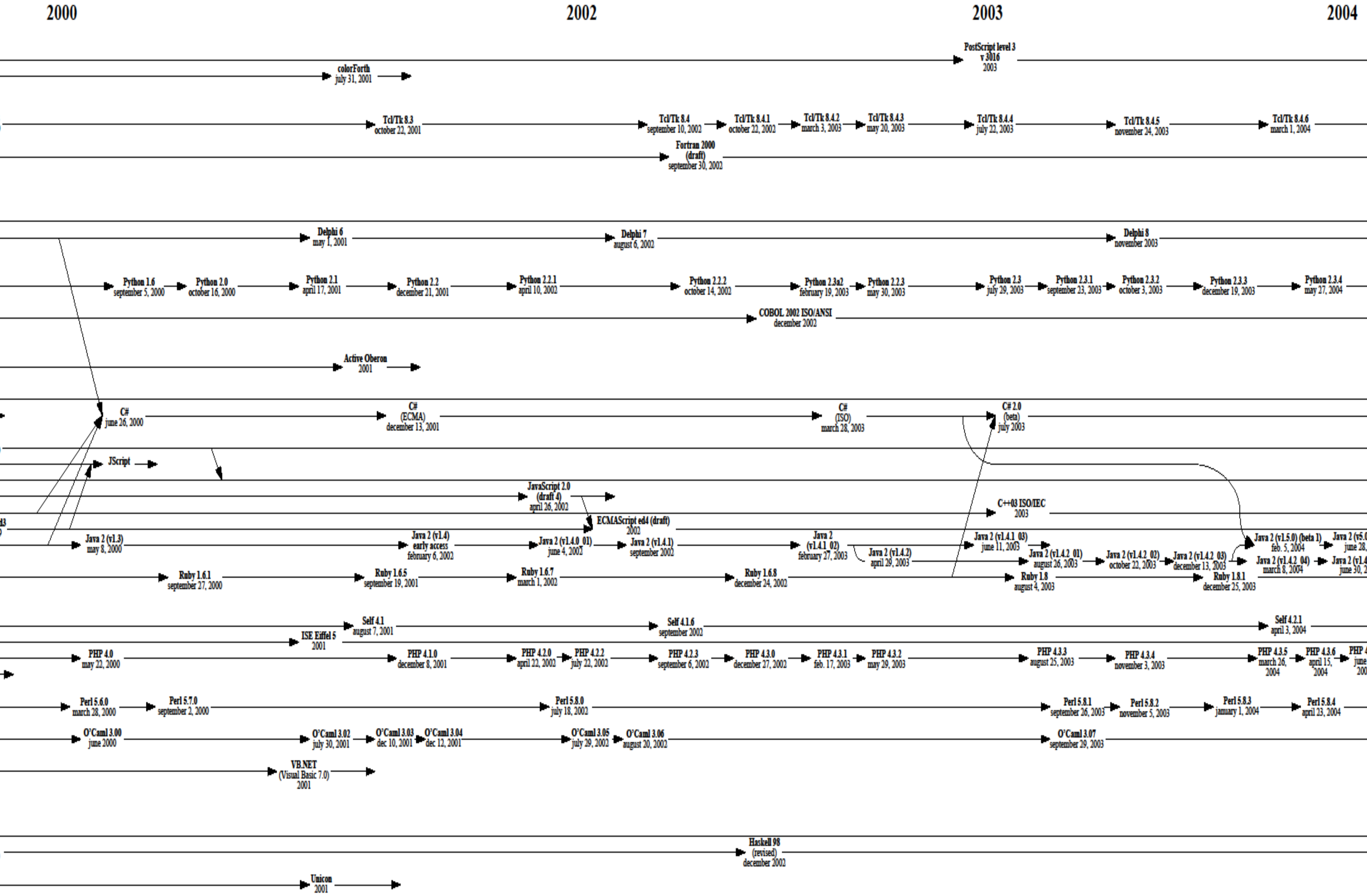
Common Lisp 1984 → Clos 1989; Common Lisp ANSI december 8, 1994

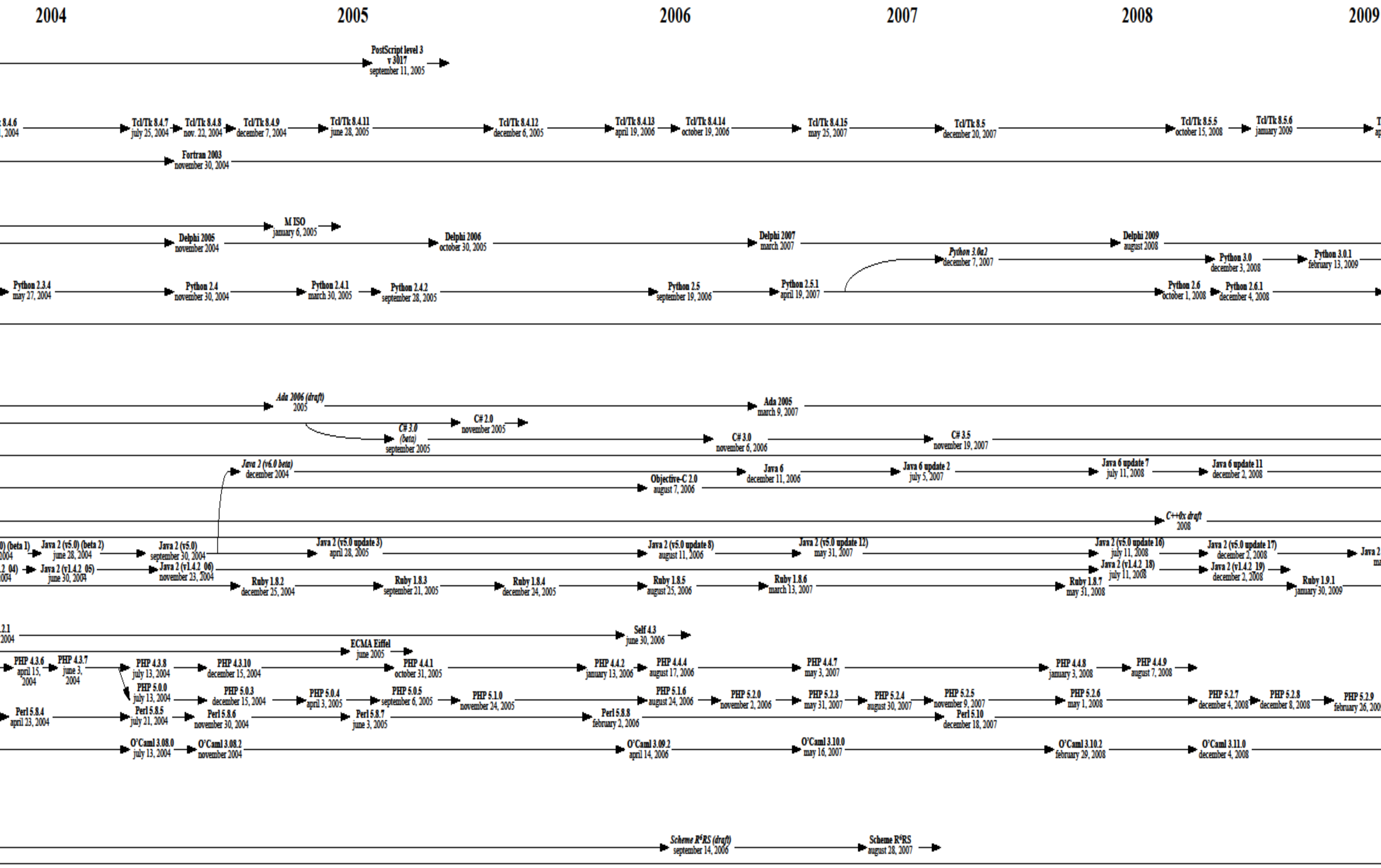
Scheme 84 1984 → Scheme IEEE 1990 → Scheme R^{5}RS 1998

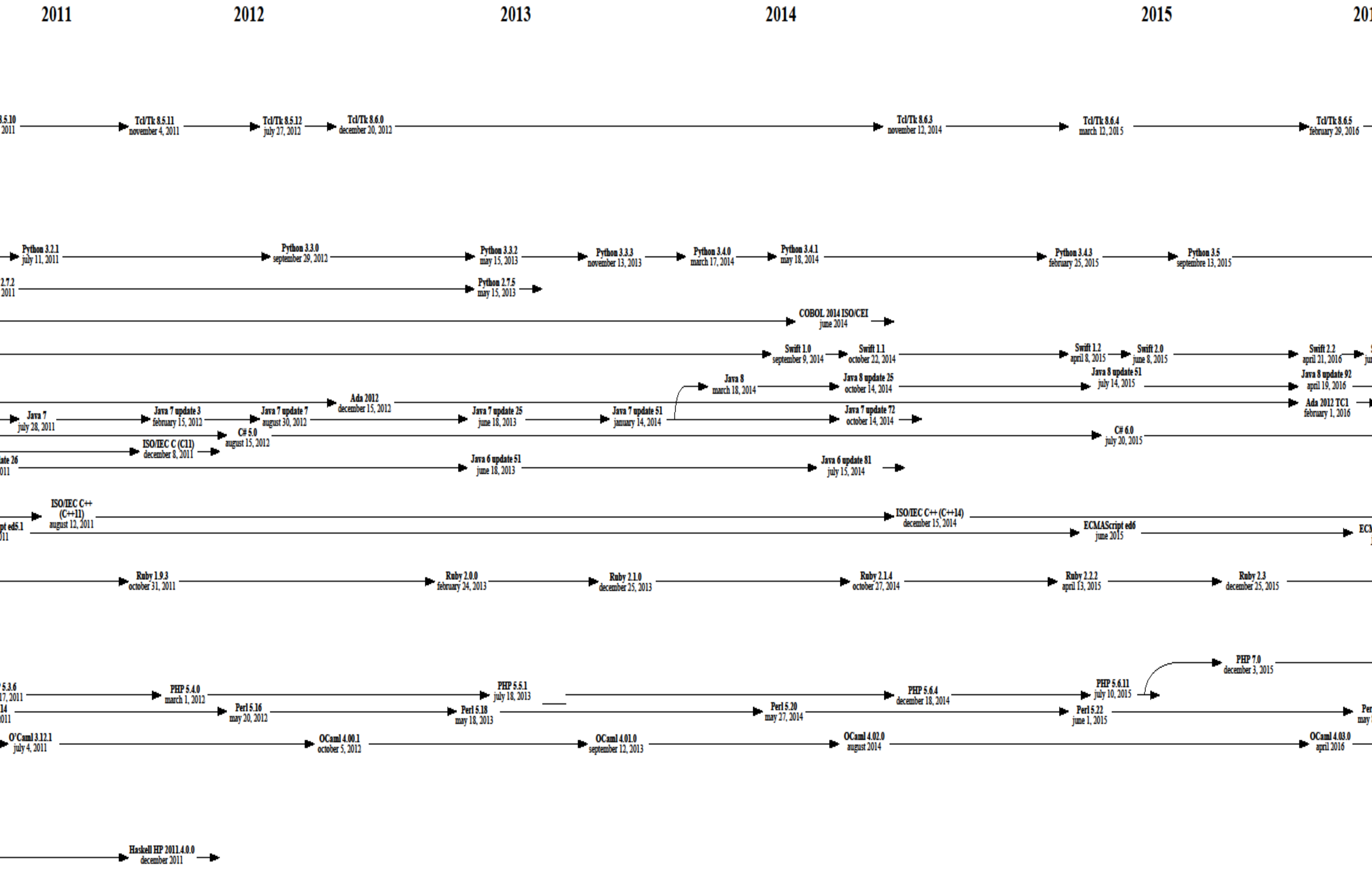
Haskell 1.0 1987 → Haskell 1.1 april 1, 1990 → Haskell 1.2 march 1992 → Haskell 1.3 may 1996 → Haskell 1.4 april 1997 → Haskell 98 february 1999

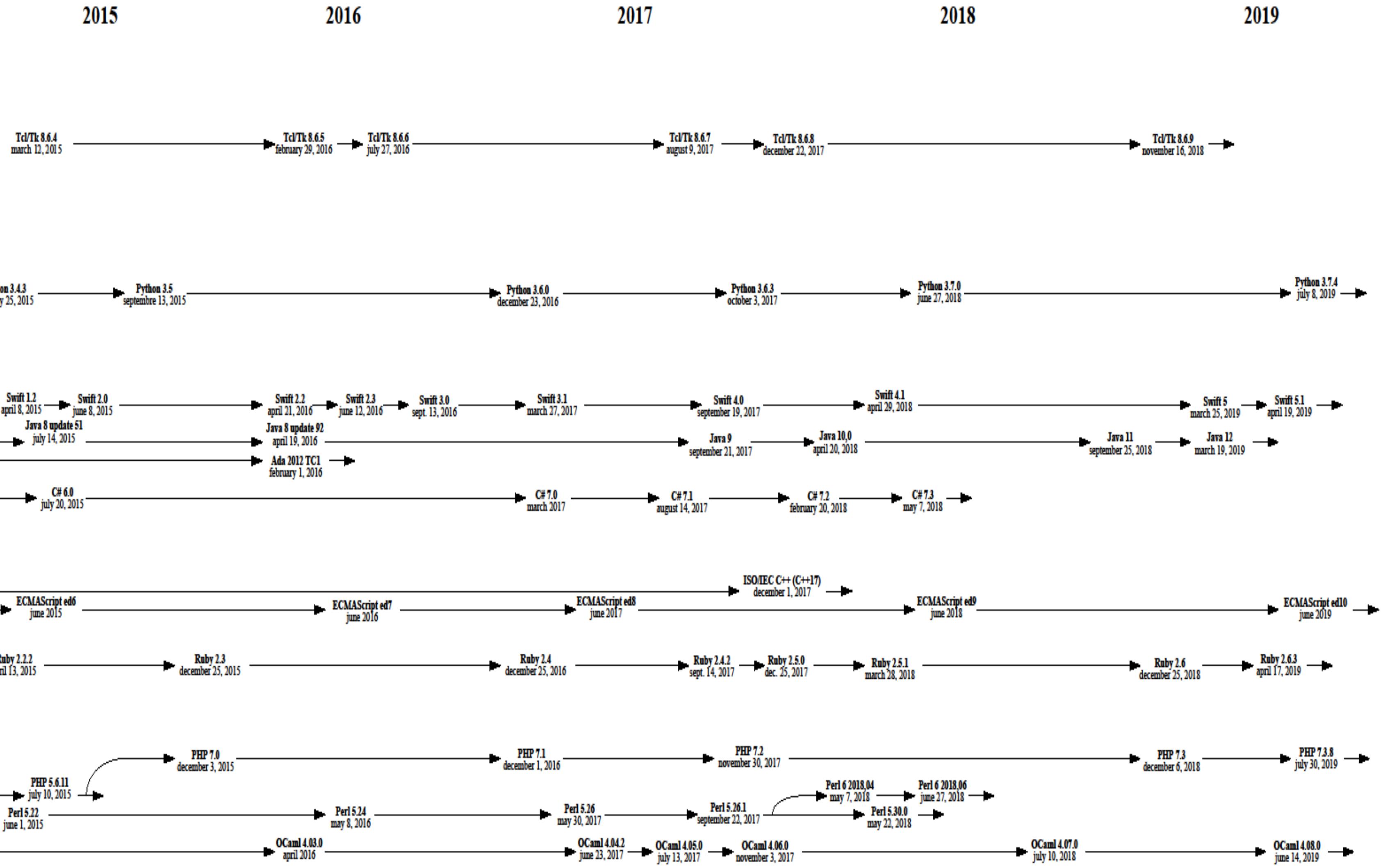
SML 1984 → SML '90 1990 → SML '97 1997

Caml 1987 → Caml 2-6.1 1991 → Caml 3.1 1993


2000
2002
2003
2004
PostScript level 3 v 3016 2003
colorForth july 31, 2001
Tcl/Tk 8.3 october 22, 2001
Tcl/Tk 8.4 september 10, 2002
Tcl/Tk 8.4.1 october 22, 2002
Tcl/Tk 8.4.2 march 3, 2003
Tcl/Tk 8.4.3 may 20, 2003
Tcl/Tk 8.4.4 july 22, 2003
Tcl/Tk 8.4.5 november 24, 2003
Tcl/Tk 8.4.6 march 1, 2004
Fortran 2000 (draft) september 30, 2002
Delphi 6 may 1, 2001
Delphi 7 august 6, 2002
Delphi 8 november 2003
Python 1.6 september 5, 2000
Python 2.0 october 16, 2000
Python 2.1 april 17, 2001
Python 2.2 december 21, 2001
Python 2.2.1 april 10, 2002
Python 2.2.2 october 14, 2002
Python 2.3a2 february 19, 2003
Python 2.2.3 may 30, 2003
Python 2.3 july 29, 2003
Python 2.3.1 september 23, 2003
Python 2.3.2 october 3, 2003
Python 2.3.3 december 19, 2003
Python 2.3.4 may 27, 2004
COBOL 2002 ISO/ANSI december 2002
Active Oberon 2001
C# june 26, 2000
C# (ECMA) december 13, 2001
C# (ISO) march 28, 2003
C# 2.0 (beta) july 2003
JScript
JavaScript 2.0 (draft 4) april 26, 2002
C++03 ISO/IEC 2003
ECMAScript ed4 (draft) 2002
Java 2 (v1.3) may 8, 2000
Java 2 (v1.4) early access february 6, 2002
Java 2 (v1.4.0_01) june 4, 2002
Java 2 (v1.4.1) september 2002
Java 2 (v1.4.1_02) february 27, 2003
Java 2 (v1.4.2) april 29, 2003
Java 2 (v1.4.1_03) june 11, 2003
Java 2 (v1.4.2_01) august 26, 2003
Java 2 (v1.4.2_02) october 22, 2003
Java 2 (v1.4.2_03) december 13, 2003
Java 2 (v1.5.0) (beta 1) feb. 5, 2004
Java 2 (v5.0 june 28,
Java 2 (v1.4.2_04) march 8, 2004
Java 2 (v1.4 june 30, 2
Ruby 1.6.1 september 27, 2000
Ruby 1.6.5 september 19, 2001
Ruby 1.6.7 march 1, 2002
Ruby 1.6.8 december 24, 2002
Ruby 1.8 august 4, 2003
Ruby 1.8.1 december 25, 2003
Self 4.1 august 7, 2001
Self 4.1.6 september 2002
Self 4.2.1 april 3, 2004
ISE Eiffel 5 2001
PHP 4.0 may 22, 2000
PHP 4.1.0 december 8, 2001
PHP 4.2.0 april 22, 2002
PHP 4.2.2 july 22, 2002
PHP 4.2.3 september 6, 2002
PHP 4.3.0 december 27, 2002
PHP 4.3.1 feb. 17, 2003
PHP 4.3.2 may 29, 2003
PHP 4.3.3 august 25, 2003
PHP 4.3.4 november 3, 2003
PHP 4.3.5 march 26, 2004
PHP 4.3.6 april 15, 2004
PHP 4 june 200
Perl 5.6.0 march 28, 2000
Perl 5.7.0 september 2, 2000
Perl 5.8.0 july 18, 2002
Perl 5.8.1 september 26, 2003
Perl 5.8.2 november 5, 2003
Perl 5.8.3 january 1, 2004
Perl 5.8.4 april 23, 2004
O'Caml 3.00 june 2000
O'Caml 3.02 july 30, 2001
O'Caml 3.03 dec 10, 2001
O'Caml 3.04 dec 12, 2001
O'Caml 3.05 july 29, 2002
O'Caml 3.06 august 20, 2002
O'Caml 3.07 september 29, 2003
VB.NET (Visual Basic 7.0) 2001
Haskell 98 (revised) december 2002
Unicon 2001


2004
2005
2006
2007
2008
2009
PostScript level 3 v 3017 september 11, 2005
8.4.6 , 2004
Tcl/Tk 8.4.7 july 25, 2004
Tcl/Tk 8.4.8 nov. 22, 2004
Tcl/Tk 8.4.9 december 7, 2004
Tcl/Tk 8.4.11 june 28, 2005
Tcl/Tk 8.4.12 december 6, 2005
Tcl/Tk 8.4.13 april 19, 2006
Tcl/Tk 8.4.14 october 19, 2006
Tcl/Tk 8.4.15 may 25, 2007
Tcl/Tk 8.5 december 20, 2007
Tcl/Tk 8.5.5 october 15, 2008
Tcl/Tk 8.5.6 january 2009
T ap
Fortran 2003 november 30, 2004
M ISO january 6, 2005
Delphi 2005 november 2004
Delphi 2006 october 30, 2005
Delphi 2007 march 2007
Delphi 2009 august 2008
Python 3.0a2 december 7, 2007
Python 3.0 december 3, 2008
Python 3.0.1 february 13, 2009
Python 2.3.4 may 27, 2004
Python 2.4 november 30, 2004
Python 2.4.1 march 30, 2005
Python 2.4.2 september 28, 2005
Python 2.5 september 19, 2006
Python 2.5.1 april 19, 2007
Python 2.6 october 1, 2008
Python 2.6.1 december 4, 2008
Ada 2006 (draft) 2005
Ada 2005 march 9, 2007
C# 2.0 november 2005
C# 3.0 (beta) september 2005
C# 3.0 november 6, 2006
C# 3.5 november 19, 2007
Java 2 (v6.0 beta) december 2004
Java 6 december 11, 2006
Java 6 update 2 july 5, 2007
Java 6 update 7 july 11, 2008
Java 6 update 11 december 2, 2008
Objective-C 2.0 august 7, 2006
C++0x draft 2008
0) (beta 1) 2004
Java 2 (v5.0) (beta 2) june 28, 2004
Java 2 (v5.0) september 30, 2004
Java 2 (v5.0 update 3) april 28, 2005
Java 2 (v5.0 update 8) august 11, 2006
Java 2 (v5.0 update 12) may 31, 2007
Java 2 (v5.0 update 16) july 11, 2008
Java 2 (v5.0 update 17) december 2, 2008
Java 2 ma
.2_04) 004
Java 2 (v1.4.2_05) june 30, 2004
Java 2 (v1.4.2_06) november 23, 2004
Java 2 (v1.4.2_18) july 11, 2008
Java 2 (v1.4.2_19) december 2, 2008
Ruby 1.8.2 december 25, 2004
Ruby 1.8.3 september 21, 2005
Ruby 1.8.4 december 24, 2005
Ruby 1.8.5 august 25, 2006
Ruby 1.8.6 march 13, 2007
Ruby 1.8.7 may 31, 2008
Ruby 1.9.1 january 30, 2009
.2.1 2004
Self 4.3 june 30, 2006
ECMA Eiffel june 2005
PHP 4.3.6 april 15, 2004
PHP 4.3.7 june 3, 2004
PHP 4.3.8 july 13, 2004
PHP 4.3.10 december 15, 2004
PHP 4.4.1 october 31, 2005
PHP 4.4.2 january 13, 2006
PHP 4.4.4 august 17, 2006
PHP 4.4.7 may 3, 2007
PHP 4.4.8 january 3, 2008
PHP 4.4.9 august 7, 2008
PHP 5.0.0 july 13, 2004
PHP 5.0.3 december 15, 2004
PHP 5.0.4 april 3, 2005
PHP 5.0.5 september 6, 2005
PHP 5.1.0 november 24, 2005
PHP 5.1.6 august 24, 2006
PHP 5.2.0 november 2, 2006
PHP 5.2.3 may 31, 2007
PHP 5.2.4 august 30, 2007
PHP 5.2.5 november 9, 2007
PHP 5.2.6 may 1, 2008
PHP 5.2.7 december 4, 2008
PHP 5.2.8 december 8, 2008
PHP 5.2.9 february 26, 200
Perl 5.8.4 april 23, 2004
Perl 5.8.5 july 21, 2004
Perl 5.8.6 november 30, 2004
Perl 5.8.7 june 3, 2005
Perl 5.8.8 february 2, 2006
Perl 5.10 december 18, 2007
O'Caml 3.08.0 july 13, 2004
O'Caml 3.08.2 november 2004
O'Caml 3.09.2 april 14, 2006
O'Caml 3.10.0 may 16, 2007
O'Caml 3.10.2 february 29, 2008
O'Caml 3.11.0 december 4, 2008
Scheme R⁶RS (draft) september 14, 2006
Scheme R⁶RS august 28, 2007


2011
2012
2013
2014
2015
201
3.5.10
2011
Tcl/Tk 8.5.11
november 4, 2011
Tcl/Tk 8.5.12
july 27, 2012
Tcl/Tk 8.6.0
december 20, 2012
Tcl/Tk 8.6.3
november 12, 2014
Tcl/Tk 8.6.4
march 12, 2015
Tcl/Tk 8.6.5
february 29, 2016
Python 3.2.1
july 11, 2011
Python 3.3.0
september 29, 2012
Python 3.3.2
may 15, 2013
Python 3.3.3
november 13, 2013
Python 3.4.0
march 17, 2014
Python 3.4.1
may 18, 2014
Python 3.4.3
february 25, 2015
Python 3.5
septembre 13, 2015
2.7.2
2011
Python 2.7.5
may 15, 2013
COBOL 2014 ISO/CEI
june 2014
Swift 1.0
september 9, 2014
Swift 1.1
october 22, 2014
Swift 1.2
april 8, 2015
Swift 2.0
june 8, 2015
Swift 2.2
april 21, 2016
Java 8
march 18, 2014
Java 8 update 25
october 14, 2014
Java 8 update 51
july 14, 2015
Java 8 update 92
april 19, 2016
Ada 2012
december 15, 2012
Ada 2012 TC1
february 1, 2016
Java 7
july 28, 2011
Java 7 update 3
february 15, 2012
Java 7 update 7
august 30, 2012
Java 7 update 25
june 18, 2013
Java 7 update 51
january 14, 2014
Java 7 update 72
october 14, 2014
C# 5.0
august 15, 2012
C# 6.0
july 20, 2015
ISO/IEC C (C11)
december 8, 2011
ate 26
011
Java 6 update 51
june 18, 2013
Java 6 update 81
july 15, 2014
ISO/IEC C++
(C++11)
august 12, 2011
ISO/IEC C++ (C++14)
december 15, 2014
pt ed5.1
011
ECMAScript ed6
june 2015
ECM
Ruby 1.9.3
october 31, 2011
Ruby 2.0.0
february 24, 2013
Ruby 2.1.0
december 25, 2013
Ruby 2.1.4
october 27, 2014
Ruby 2.2.2
april 13, 2015
Ruby 2.3
december 25, 2015
PHP 7.0
december 3, 2015
5.3.6
17, 2011
PHP 5.4.0
march 1, 2012
PHP 5.5.1
july 18, 2013
PHP 5.6.4
december 18, 2014
PHP 5.6.11
july 10, 2015
14
011
Perl 5.16
may 20, 2012
Perl 5.18
may 18, 2013
Perl 5.20
may 27, 2014
Perl 5.22
june 1, 2015
Per
may
O'Caml 3.12.1
july 4, 2011
OCaml 4.00.1
october 5, 2012
OCaml 4.01.0
september 12, 2013
OCaml 4.02.0
august 2014
OCaml 4.03.0
april 2016
Haskell HP 2011.4.0.0
december 2011


2015
2016
2017
2018
2019
Tcl/Tk 8.6.4
march 12, 2015
Tcl/Tk 8.6.5
february 29, 2016
Tcl/Tk 8.6.6
july 27, 2016
Tcl/Tk 8.6.7
august 9, 2017
Tcl/Tk 8.6.8
december 22, 2017
Tcl/Tk 8.6.9
november 16, 2018
on 3.4.3
y 25, 2015
Python 3.5
septembre 13, 2015
Python 3.6.0
december 23, 2016
Python 3.6.3
october 3, 2017
Python 3.7.0
june 27, 2018
Python 3.7.4
july 8, 2019
Swift 1.2
april 8, 2015
Swift 2.0
june 8, 2015
Swift 2.2
april 21, 2016
Swift 2.3
june 12, 2016
Swift 3.0
sept. 13, 2016
Swift 3.1
march 27, 2017
Swift 4.0
september 19, 2017
Swift 4.1
april 29, 2018
Swift 5
march 25, 2019
Swift 5.1
april 19, 2019
Java 8 update 51
july 14, 2015
Java 8 update 92
april 19, 2016
Java 9
september 21, 2017
Java 10,0
april 20, 2018
Java 11
september 25, 2018
Java 12
march 19, 2019
Ada 2012 TC1
february 1, 2016
C# 6.0
july 20, 2015
C# 7.0
march 2017
C# 7.1
august 14, 2017
C# 7.2
february 20, 2018
C# 7.3
may 7, 2018
ISO/IEC C++ (C++17)
december 1, 2017
ECMAScript ed6
june 2015
ECMAScript ed7
june 2016
ECMAScript ed8
june 2017
ECMAScript ed9
june 2018
ECMAScript ed10
june 2019
uby 2.2.2
ril 13, 2015
Ruby 2.3
december 25, 2015
Ruby 2.4
december 25, 2016
Ruby 2.4.2
sept. 14, 2017
Ruby 2.5.0
dec. 25, 2017
Ruby 2.5.1
march 28, 2018
Ruby 2.6
december 25, 2018
Ruby 2.6.3
april 17, 2019
PHP 7.0
december 3, 2015
PHP 7.1
december 1, 2016
PHP 7.2
november 30, 2017
PHP 7.3
december 6, 2018
PHP 7.3.8
july 30, 2019
PHP 5.6.11
july 10, 2015
Perl 6 2018.04
may 7, 2018
Perl 6 2018.06
june 27, 2018
Perl 5.22
june 1, 2015
Perl 5.24
may 8, 2016
Perl 5.26
may 30, 2017
Perl 5.26.1
september 22, 2017
Perl 5.30.0
may 22, 2018
OCaml 4.03.0
april 2016
OCaml 4.04.2
june 23, 2017
OCaml 4.05.0
july 13, 2017
OCaml 4.06.0
november 3, 2017
OCaml 4.07.0
july 10, 2018
OCaml 4.08.0
june 14, 2019

Programming Language History
2010s

- Multi paradigm integration, especially OO+FP(+concurrency)
  - C#, C++ and Java
  - Python
  - Ruby
  - Groovy
  - Clojure
  - Fortress
  - Scala
  - O’Caml, F#
  - Haskell
  - Erlang
  - Swift, DART, RUST, Kotlin

```
-- Hello World in Haskell

main = putStrLn "Hello World“
```

```
%% Hello World in Erlang

-module(hello).

-export([hello/0]).

hello() ->
   io:format("Hello World!~n", []).
```

```
// Hello world in Swift

println("Hello, world!")
```

```
// Hello world in Dart

main() {
   print('Hello world!');
}
```

```
// Hello world in Kotlin

fun main(args : Array<String>) {
    println("Hello, world!")
}
```

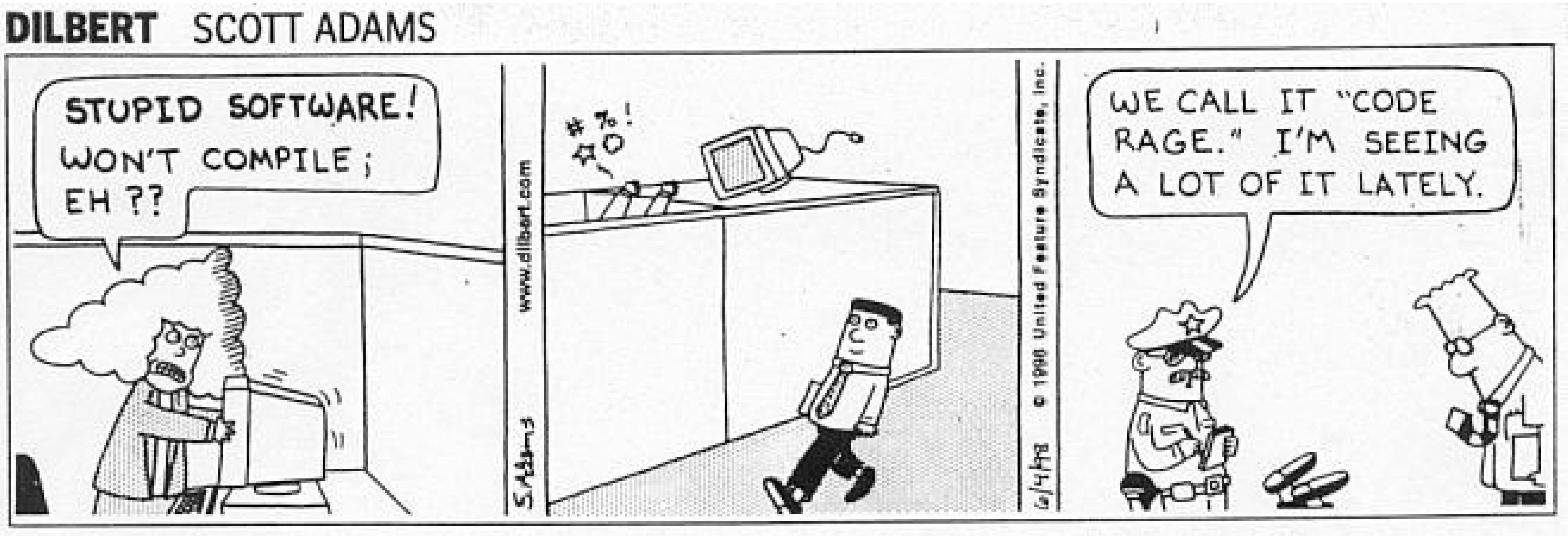
Three Trends

- Declarative programming languages in vogue again
  - Especially functional
- Dynamic Programming languages gained momentum, but …
- Concurrent Programming languages came back on the agenda
  - Reactive programming
    - (a special kind of concurrent programming)

So what can you do in your projects?

- Look at code in the languages you know
- Use Sebesta’s Language Evalualtion criteria to those languages
- Look at code in languages you do not know
- Make a list of language features you like
- Make a list of language features you dislike
- Creat some example programs

So how would you like to programme in 20 years?


DILBERT SCOTT ADAMS
STUPID SOFTWARE! WON'T COMPILE; EH ??
www.dilbert.com
S. Adams
# % !
© 1998 United Feature Syndicate, Inc.
6/4/98
WE CALL IT "CODE RAGE." I'M SEEING A LOT OF IT LATELY.