OBJECT-ORIENTED PROGRAMMING

Lecture 1: Programming Languages: History and Paradigms

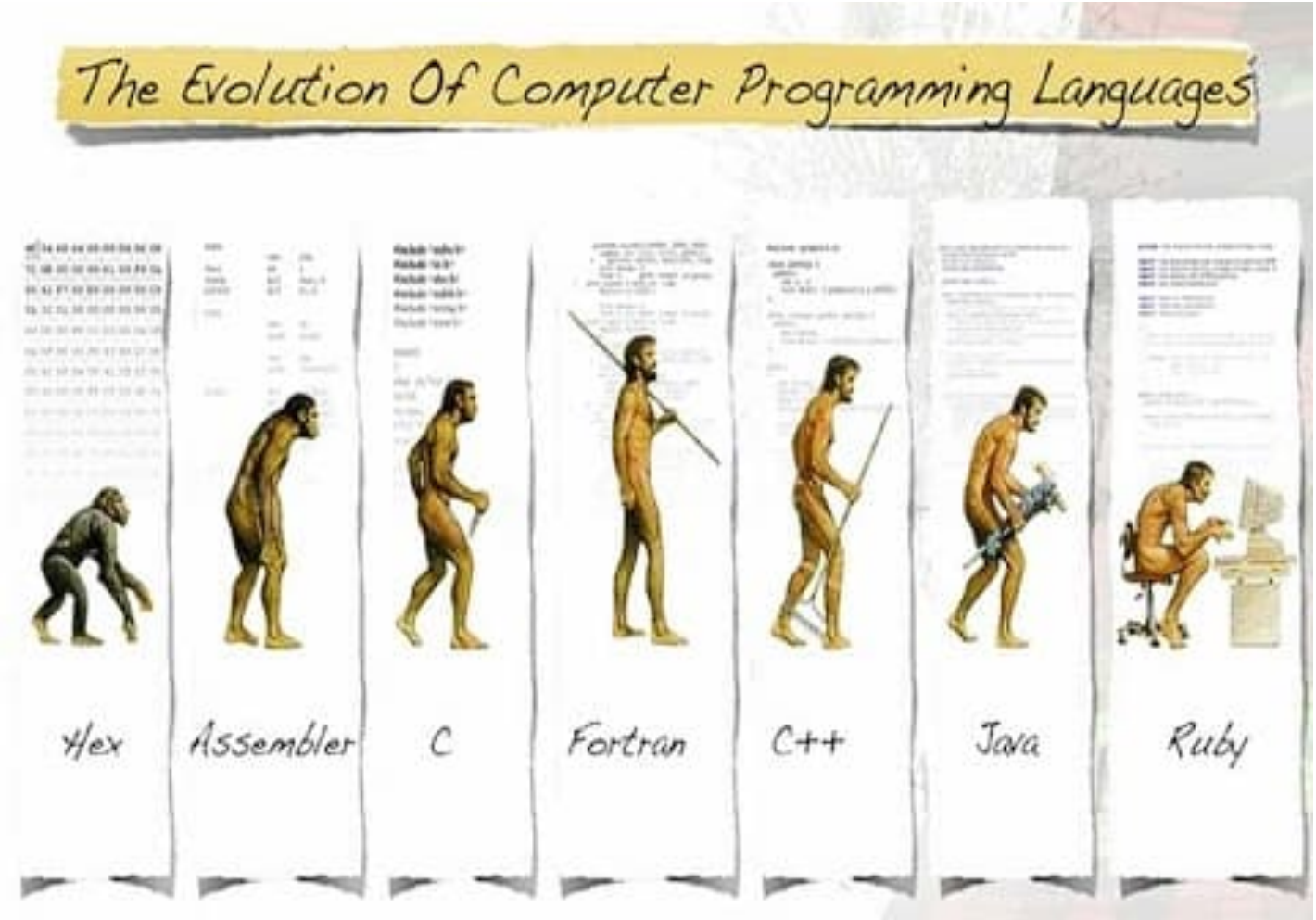
Po-Chun Huang (黃柏鈞),

Department of Electronic Engineering, Taipei Tech

Agenda

- History of computers
- History of programming languages
- Programming paradigms

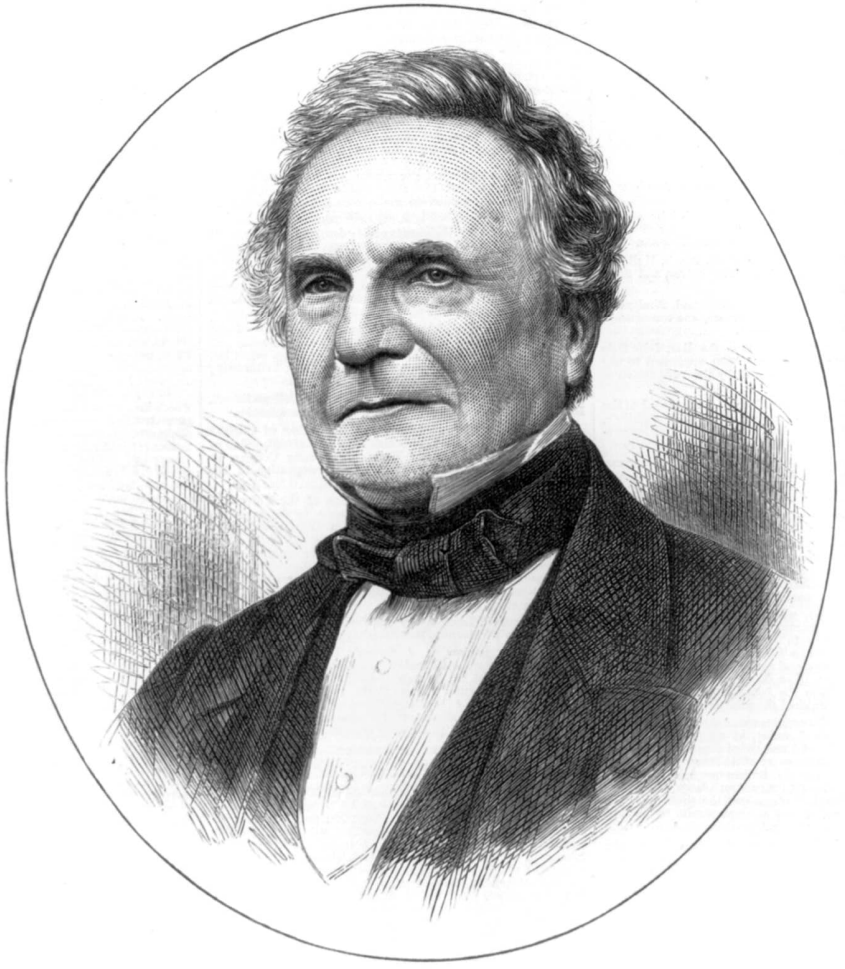
History



https://i.ytimg.com/vi/WZclSqf5_6w/hqdefault.jpg

Inventor of The First Digital Computer

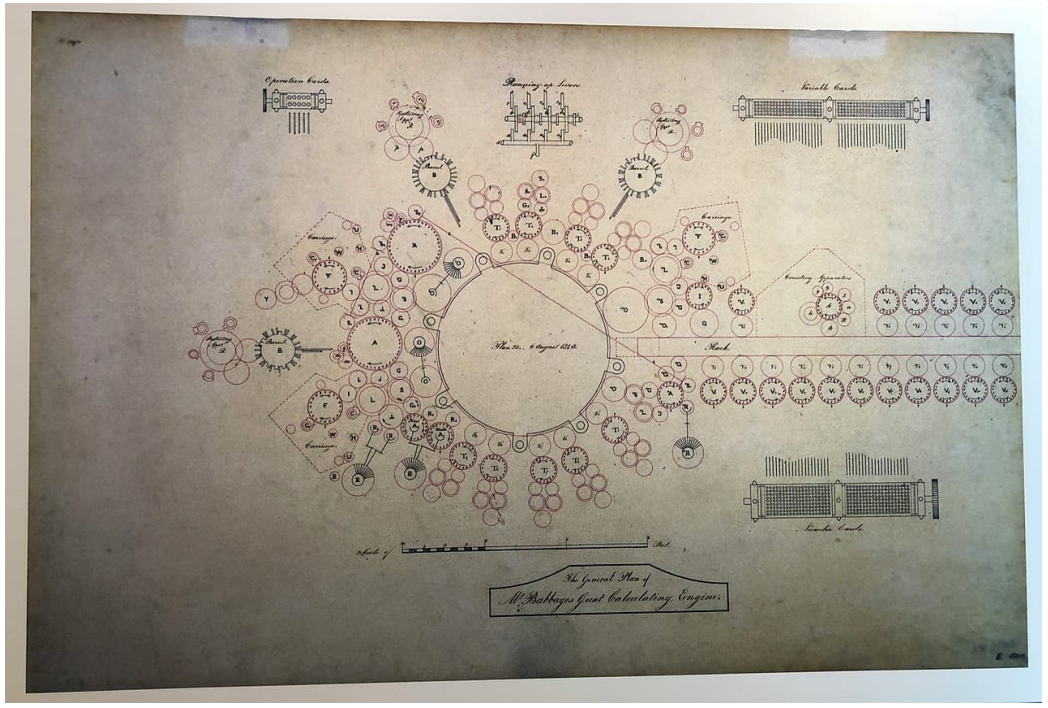
- Charles Babbage (1791–1871) invented the first **digital computer**; he is the father of computer.

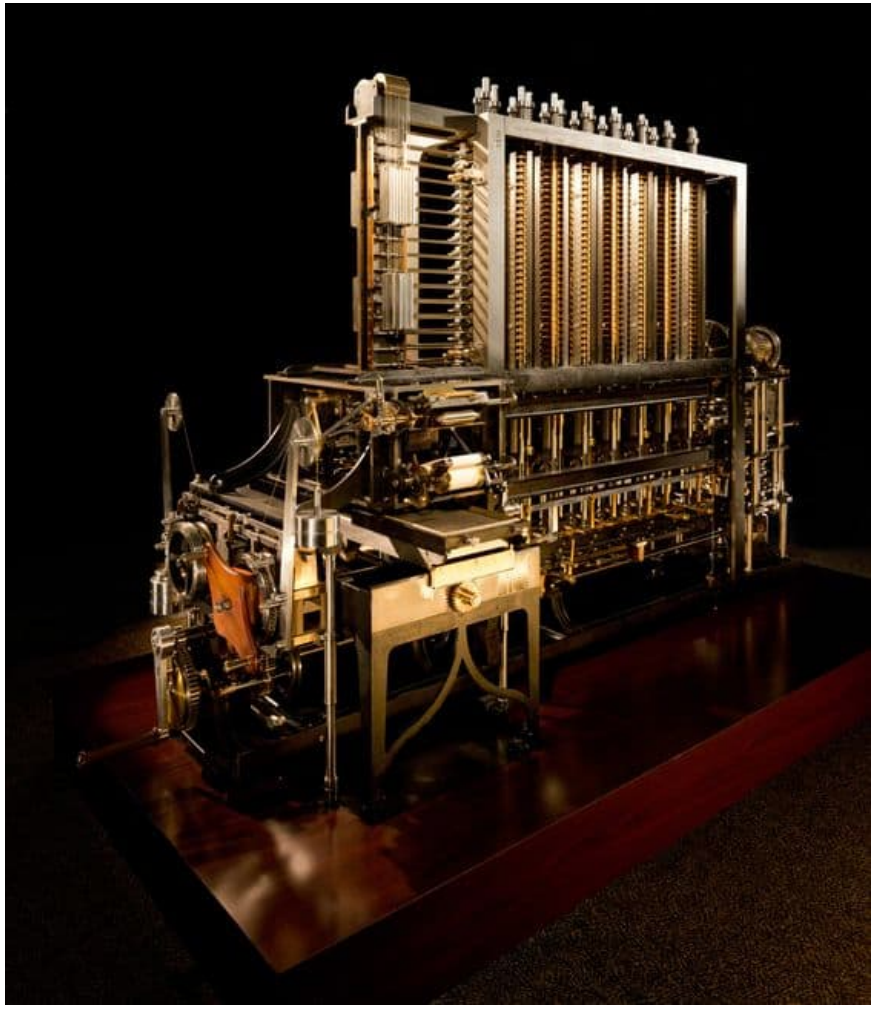


https://www.britannica.com/biography/Charles-Babbage; https://en.wikipedia.org/wiki/Charles_Babbage

The First Digital Computer

- Babbage's **analytical machine** (c. 1840) is the first digital computer in history.





https://www.pinterest.com/pin/450148925239734298/; https://en.wikipedia.org/wiki/Analytical_Engine#/media/File:Babbage_Analytical_Engine_Plan_1840_CHM.agr.jpg

How Powerful Is the First Digital Computer?

- The analytical machine:
  - Stores 1000 40-digit integers~**16.2 KB**.
  - Performs **addition**, **subtraction**, **multiplication**, **division**, **comparison**, and **square root**.
  - Uses punch cards to read in the programs.

https://en.wikipedia.org/wiki/Analytical_Engine; https://en.wikipedia.org/wiki/Analytical_Engine#/media/File:PunchedCardsAnalyticalEngine.jpg

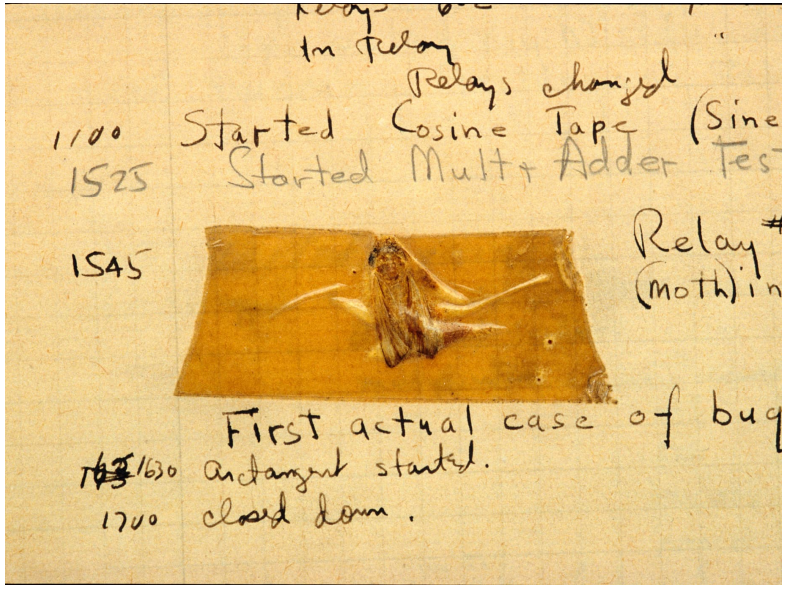
The First Programmer

- Ada Lovelace (1815–1852) is the first **programmer** in history.
- The programming language Ada is named after her.

https://www.britannica.com/biography/Ada-Lovelace

The First "Bug" in History

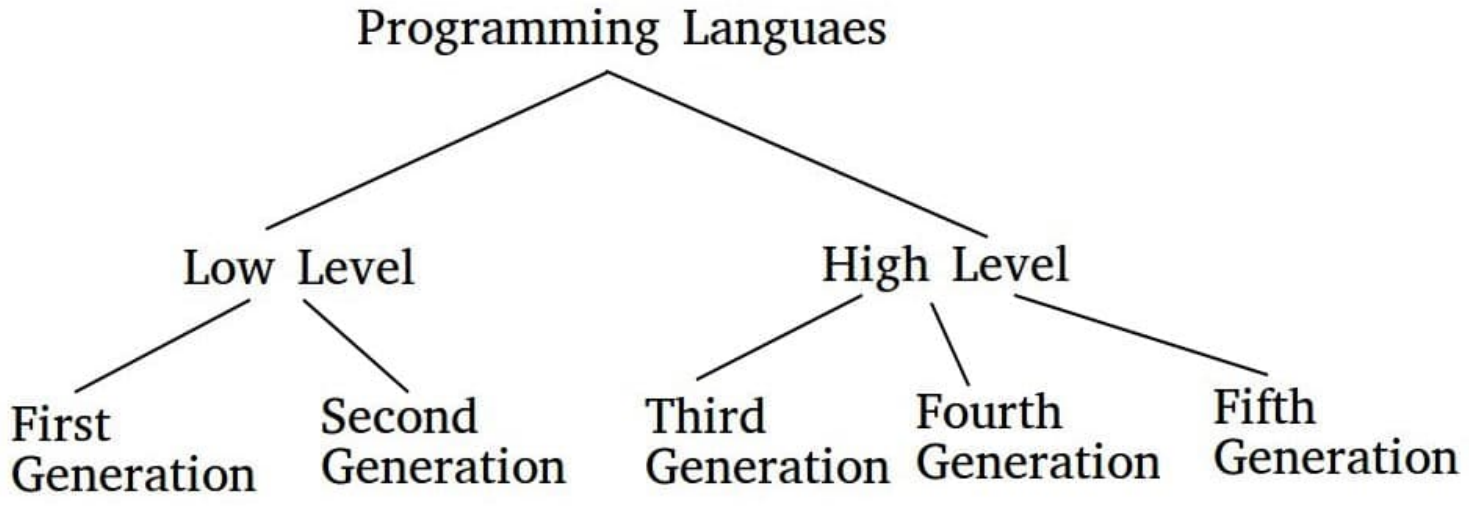
- In 1944, the rear admiral Grace Hopper (1906–1992) found the first (physical) **computer bug** in **Harvard Mark II computer** in history.



https://zh.wikipedia.org/wiki/%E8%91%9B%E9%BA%97%E7%B5%B2%C2%B7%E9%9C%8D%E6%99%AE; https://www.thoughtco.com/howard-aiken-and-grace-hopper-4078389; https://www.britannica.com/biography/Grace-Hopper

Generation of Computer Programming Languages

- **Gen 1 (low-level):** Machine language.
- **Gen 2 (low-level):** Assembly languages.
- **Gen 3 (high-level):** C, C++, Java, Visual Basic and JavaScript.
- **Gen 4 (with natural language-like syntaxes):** Perl, Python, Ruby, SQL, MatLab (Matrix Laboratory).
- **Gen 5 (with visual tools):** Mercury, OPS5, and Prolog.



https://www.geeksforgeeks.org/generation-programming-languages/

Computer Programming Paradigms

- **Imperative** 命令式程式語言 C, C++
  - Procedure
  - **Object-oriented (OO)** 結構化程式語言
- **Declarative**
  - Functional
  - Logic
  - Mathematical
  - Reactive

因為 Procedure 會有
程式碼的錯位 Redundancy
的問題，
才有物件導向的出現.

Imperative Programming

CPU有64位元，int會存成8個位元

- With **imperative programming**, a program consists of a number of **statements** that command the computer to do something.

述句

- **Examples**

character out 印出

右邊流到左邊

  - `cout << “Hello, World!\n“;`
  - `int i = 5, j = 8;` 初始化
  - `i++;` 後置遞增：把i的值讀出來加1再傳回去.
  - `i = j + 3;`
  - `int k = max( i, j );`
  - `fprintf( fp1, “Hello, World!\n" );`

函數呼叫

Functional Programming (FP)

函數式程式語言

- In contrast to imperative programming, **functional programming (FP)** tells the computer what kinds of solutions are needed.
- Programming languages that support FP: Haskell, Scheme, ML, Ocaml, Scala, Erlang, LISP, R, and Mathematica.
- **Example**
  - **Imperative version**

```
a = 0
def increment():
    global a
    a += 1
```

  - **Functional version**

```
def increment(a):
    return a + 1
```

輸入 a.
a+1 後回傳

https://maryrosecook.com/blog/post/a-practical-introduction-to-functional-programming

Structural Programming

- **Structural programming** splits large programs into functions, code blocks, `while` and `for` structures to avoid **spaghetti code**.

- Spaghetti code for computing squares

```
int i = 0;
REPEAT: i++;
cout << i << " sqr = " << i*i;
if( i >= 10 ) goto DONE;
goto REPEAT;
DONE: cout << "Done.\n";
```

- Better code

```
for( i = 1; i < 10; i++ )
    cout << i << " sqr = " << i*i;
cout << " Done.\n";
```

- `Goto` statements are considered harmful. Really?

https://zh.wikipedia.org/wiki/%E9%9D%A2%E6%9D%A1%E5%BC%8F%E4%BB%A3%E7%A0%81

Structural Programming (Cont'd)

- Structures in structural programming
  - **Sequence**: normal statements ran one by one.
  - **Selection**
    - `if…then…else`
    - `switch…case`
    - `?:`
  - **Repetition**
    - `for`
    - `while…`
    - `do…while…`

組譯 答、

- Few languages w/o structures are called **non-structured programming**, e.g., machine languages, COBOL, and early assembly languages.

Is Goto Considered Harmful?

- Goto can help jump out from deeply-nested for or while structures.

```
for( int i = 0; i < 1000; i++ ) {
    for( int j = 0; j < 500; j++ ) {
        goto EXIT;
    }
}
EXIT: cout << result << endl;
```

- Extensive reading: https://homepages.cwi.nl/~storm/teaching/reader/Dijkstra68.pdf

Representative Programming Languages

- **High-level languages**: Java, Python, C#, Visual Basic, JavaScript, PHP, SQL, R, Groovy, Go, Ruby, Swift, MATLAB, Perl, and Objective-C
- **Mid-level languages**: C and C++
- **Low-level languages**: assembly language & machine language

高階：好的表達力

低階：

TIOBE Index of Programming Languages

Feb 2023	Feb 2022	Change	Programming Language	Ratings	Change
1	1		Python	15.49%	+0.16%
2	2		C	15.39%	+1.31%
3	4	↑	C++	13.94%	+5.93%
4	3	↓	Java	13.21%	+1.07%
5	5		C#	6.38%	+1.01%
6	6		Visual Basic	4.14%	-1.09%
7	7		JavaScript	2.52%	+0.70%
8	10	↑	SQL	2.12%	+0.58%
9	9		Assembly language	1.38%	-0.21%
10	8	↓	PHP	1.29%	-0.49%
11	11		Go	1.11%	-0.12%

https://www.tiobe.com/tiobe-index/

We Use C++ in This Course. But Why?

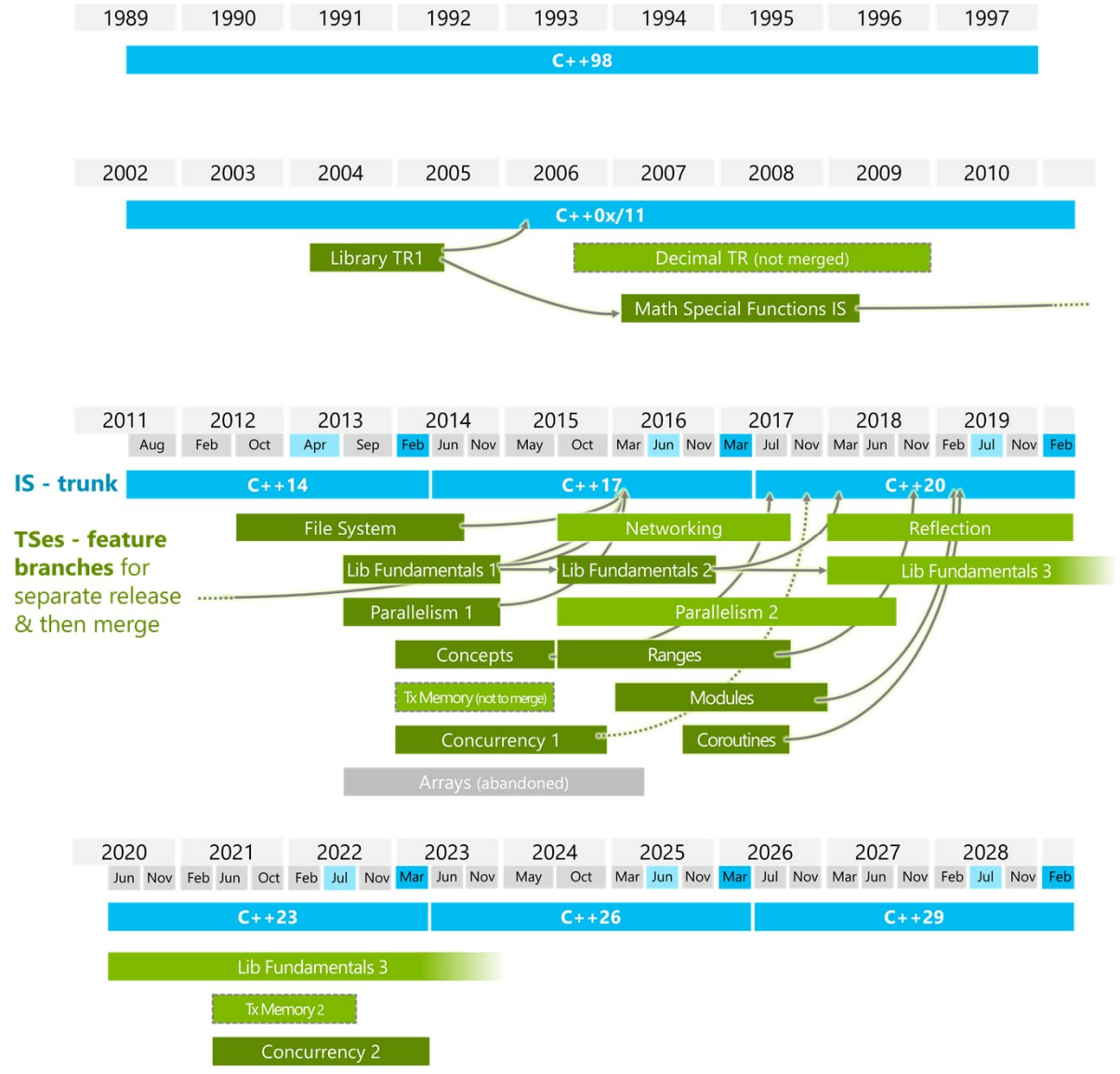
isocpp.org

- C++ has the **best popularity** (if calculated together with its predecessor, C).
- C++ provides more **direct control of computer hardware**.
- C++ is well defined by the advancing **ISO standards**.
- C++ has an **open community**, not owned by proprietary companies.
- C++ has many free and excellent **software tools**.
- Abundant **online resources** for troubleshooting.
- **Python** has a similar (and simpler) syntax to C++.

History of C++

- Combined programming language (CPL) (1963) ⇒
  Basic CPL (BCPL) (1967) ⇒
  B (1969) ⇒
  C (1972) ⇒
  C++ (1983)
- Although C# seems C++++, it is *not* a successor of C++.

C++ Is an Ever-evolving Language



https://isocpp.org/files/img/timeline-2022-07.png

Code Segment for “Max” in CPL

```
Max(Items, ValueFunction) = value of
§ (Best, BestVal) = (NIL, -∞)
while Items do §
(Item, Val) = (Head(Items),
ValueFunction(Head(Items)))
if Val > BestVal then (Best, BestVal) := (Item,
Val)
Items := Rest(Items) §
result is Best §
```


https://en.wikipedia.org/wiki/CPL_(programming_language)

Code Segment for “Factorial” in BCPL

```
GET "LIBHDR"
LET START() = VALOF $(
    FOR I = 1 TO 5 DO
        WRITEF("%N! = %I4*N", I, FACT(I))
    RESULTIS 0
$)
AND FACT(N) = N = 0 -> 1, N * FACT(N - 1)
```


https://en.wikipedia.org/wiki/BCPL

Code Segment for “Base Conversion” in B

```
printn(n, b) {
    extrn putchar;
    auto a;
    if (a = n / b) /* assignment, not comparison */
        printn(a, b); /* recursive */
    putchar(n % b + '0');
}
```


https://en.wikipedia.org/wiki/B_(programming_language)

Code Segment for “Fibonacci Series” in C

```
#include <stdio.h>
int main() {
    int i, n, t1 = 0, t2 = 1, nextTerm;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci Series: ");
    for (i = 1; i <= n; ++i) {
        printf("%d, ", t1);
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
    }
    return 0;
}
```


https://www.programiz.com/c-programming/examples/fibonacci-series

Imperative Programming: A Closer Look

- Program is realized by statements.
- Each statement takes some actions.
- A program may run different statements according to the cases, i.e., have multiple **program flows**.
- A statement can be interpreted into one or more **instructions** of target CPU.

Procedural Programming: A Closer Look

- Managing a large number of statements is complex.
- **Procedural programming** groups multiple statements into **procedures** to make the program logic clear.
- Procedures can be called **iteratively** or **recursively** to realize some amazing tricks…

Thank You Very Much!

Q&A?