Introduction

FRA 142 Computer Programming for Robotics and Automation Engineering II

Suriya Natsupakpong, PhD

Institute of Field Robotics (FIBO)

King Mongkut's University of Technology Thonburi (KMUTT)

Learning Outcome for FRA 142

เมื่อนักศึกษาได้เรียนรายวิชานี้จบแล้ว นักศึกษาสามารถ

1. ผู้เรียนสามารถประยุกต์ใช้หลักการ เทคนิค และการเขียนโปรแกรมแบบเชิงออบเจ็ค
    - เขียนโปรแกรมโดยใช้ภาษา C/C++ ได้
    - ออกแบบและเขียนโปรแกรมแบบเชิงวัตถุได้ ในภาษาต่าง ๆ
    - ประยุกต์ใช้งานคุณสมบัติของ OOP ได้ (Encapsulation, Inheritance, Polymorphism)
2. ผู้เรียนสามารถประยุกต์ใช้ทฤษฎีในการพัฒนาซอฟต์แวร์
3. ผู้เรียนมีทักษะการพัฒนาซอฟต์แวร์ที่ซับซ้อน
4. ผู้เรียนสามารถพัฒนางานซอฟต์แวร์ร่วมกับผู้อื่นได้
5. ผู้เรียนนำความรู้ที่ได้ไปประยุกต์ใช้งานจริงกับปัญหาในชีวิตประจำวันได้

Course Schedule

@ ห้องคอมพิวเตอร์ FB306

วันอังคาร

Sec 1 8:45 - 10:45 น.

Sec 2 10:45 - 12:45 น.

วันพฤหัสบดี

Sec 1 13:30 - 15:30 น.

Sec 2 15:30 - 17:30 น.

Week	Date	Topics
1	1/17	Class Introduction
	1/19	Lab 0: Preparation
2	1/24	Basic C++ Sytax: variable types, if-else, loop
	1/26	Lab 1: Basic C++ Programming
3	1/31	Pointer, References
	2/2	Lab 2: Pointer Manipulation
4	2/7	OOP (1): Abstraction, Encapsulation, Class
	2/9	Lab 3: Data Structure (1) - Node and Linked List
5	2/14	OOP (2): Inheritance
	2/16	Lab 4: Data Structure (2) - Stack and Queue
6	2/21	Exam Break
	2/23	Exam Break
7	2/28	OOP (3): Polymorphism
	3/2	Lab 5: Data Structure (3) - Sorting Algorithm
8	3/7	Software Dev (1)
	3/9	Lab 6: Version Control and Git Introduction
9	3/14	Software Dev (2)
	3/16	Lab 7: Building User Stories and Generating Requirements

Week	Date	Topics
10	3/21	Software Dev (3)
	3/23	Lab 8: Introduction to Pygame
11	3/28	Software Dev (4)
	3/30	Lab 9: Using Pygame Events
12	4/4	Exam Break
	4/6	Exam Break
13	4/11	Songkran Break
	4/13	Songkran Break
14	4/18	
	4/20	Project Time
15	4/25	
	4/27	Project Time
16	5/2	
	5/4	Project Time
17	5/9	
	5/11	Project Time
18	5/16	
	5/18	Project Time

Textbook

+ Recommended book

P. Deitel and H. Deitel. (2013). C++ How to Program. 9th edition. Prentice-Hall.

D.S.Malik. (2013). C++ Programming: From Problem Analysis to Program Design. 6th edition. Cengage Learning.

Roger Pressman and Bruce Maxim. (2014). **Software Engineering: A Practitioner's Approach**. 8th edition. McGraw-Hill Education.

Bernd Bruegge, Allen Dutoit: (2003). **Object-Oriented Software Engineering: Using UML, Patterns, and Java**. Prentice Hall.

+ Additional books

B. Eckel. (2000). Thinking in C++: Introduction to Standard C++, Volume One. 2nd edition. Printice-Hall.

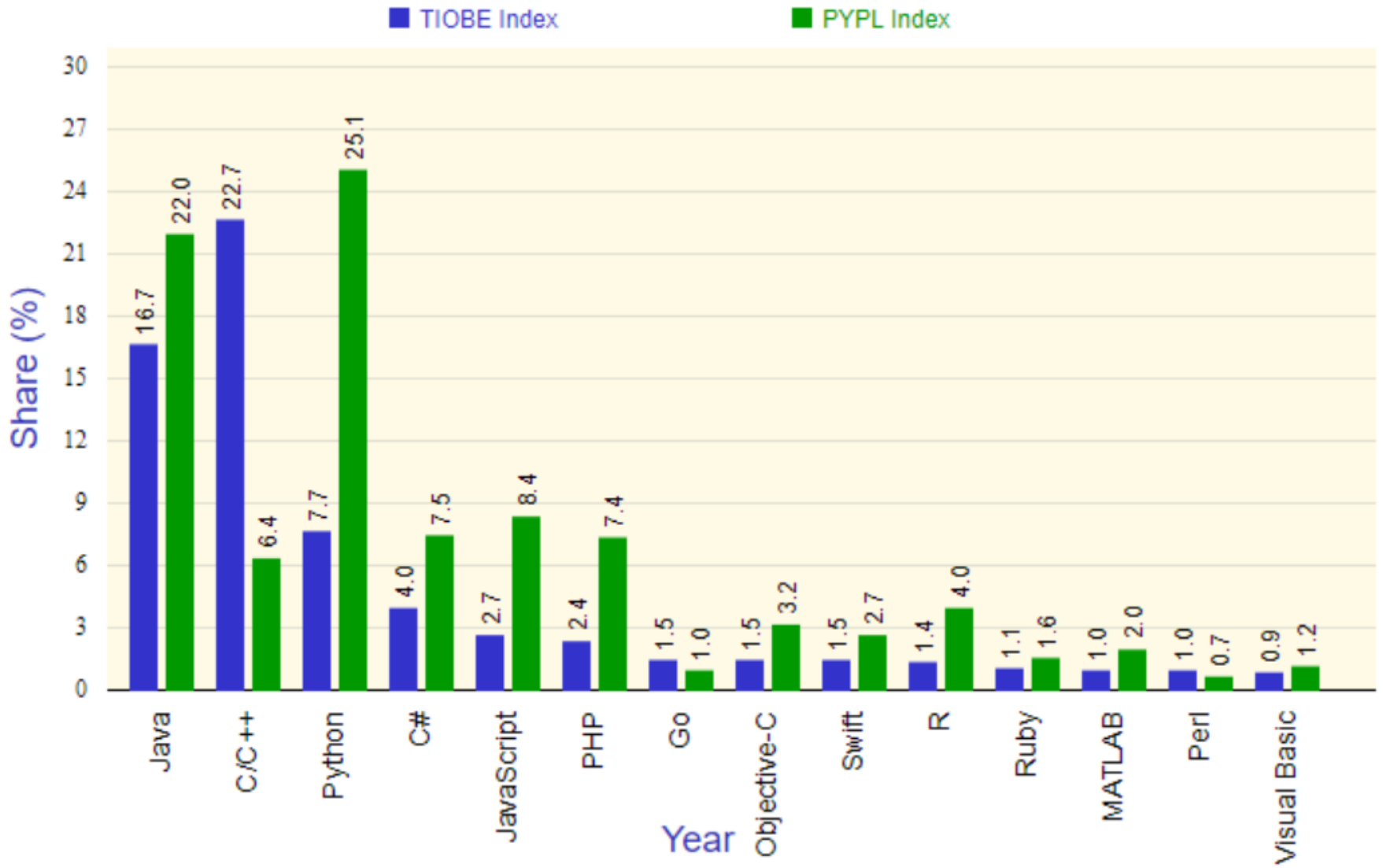
B. Eckel. (2003). Thinking in C++: Practical Programming, Volume Two. 2nd edition. Printice-Hall.

7 Secrets to Staying Motivated When Learning to Code

- Make sure this is actually something you want to do
- Start small, celebrate the little things, and build, build, build!
- Get a mentor
- Maintain a Portfolio
- Just do it. Or just do nothing
- Balance
- Be a part of a supportive community

https://www.codementor.io/learn-programming/experts-reveal-their-programming-motivation

Top Computer Language (Nov 2018)



http://statisticstimes.com/tech/top-computer-languages.php

Top Computer Language (Dec 2022)

Dec 2022	Dec 2021	Change	Programming Language	Ratings	Change
1	1		Python	16.66%	+3.76%
2	2		C	16.56%	+4.77%
3	4	↑	C++	11.94%	+4.21%
4	3	↓	Java	11.82%	+1.70%
5	5		C#	4.92%	-1.48%
6	6		Visual Basic	3.94%	-1.46%
7	7		JavaScript	3.19%	+0.90%
8	9	↑	SQL	2.22%	+0.43%
9	8	↓	Assembly language	1.87%	-0.38%
10	12	↑	PHP	1.62%	+0.12%
11	11		R	1.25%	-0.34%
12	19	⇈	Go	1.15%	+0.20%
13	13		Classic Visual Basic	1.15%	-0.13%
14	20	⇈	MATLAB	0.95%	+0.03%
15	10	⇊	Swift	0.91%	-0.86%

https://www.tiobe.com/tiobe-index/#2022

Worldwide, Jan 2023 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	27.93 %	-0.9 %
2		Java	16.78 %	-1.3 %
3		JavaScript	9.63 %	+0.5 %
4	↑	C#	6.99 %	-0.3 %
5	↓	C/C++	6.9 %	-0.5 %
6		PHP	5.29 %	-0.8 %
7		R	4.03 %	-0.2 %
8	↑↑↑	TypeScript	2.79 %	+1.0 %
9		Swift	2.23 %	+0.3 %
10	↓↓	Objective-C	2.2 %	-0.1 %
11	↑↑	Go	1.94 %	+0.7 %
12	↑↑↑	Rust	1.9 %	+0.9 %
13	↓	Kotlin	1.81 %	+0.1 %
14	↓↓↓↓	Matlab	1.63 %	-0.1 %
15	↑	Ruby	1.13 %	+0.3 %

https://pypl.github.io/PYPL.html

Why C / C++

Choose a Ranking (choose a weighting or make your own)

IEEE Spectrum | Trending | Jobs | Open | Custom

Edit Ranking | Add a Comparison |

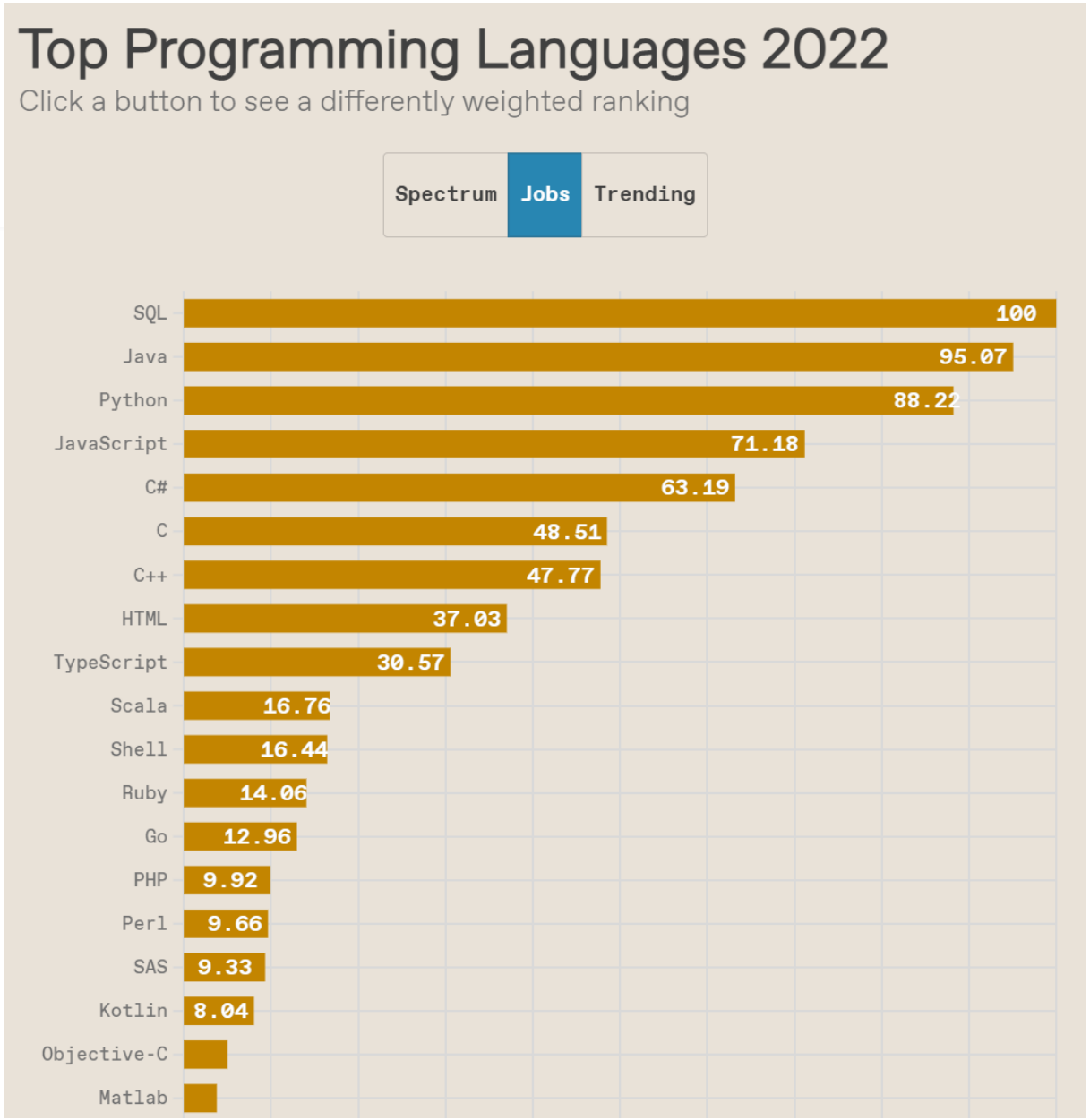
Embedded system developers

Language Types (click to hide)

Web | Mobile | Enterprise | Embedded

Language Rank	Types	Jobs Ranking
1. Python		100.0
2. Java		99.2
3. C		98.8
4. C++		94.6
5. C#		86.2
6. JavaScript		85.7
7. Assembly		83.4
8. PHP		83.1
9. HTML		81.3
10. Scala		76.5

http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018



https://spectrum.ieee.org/top-programming-languages-2022

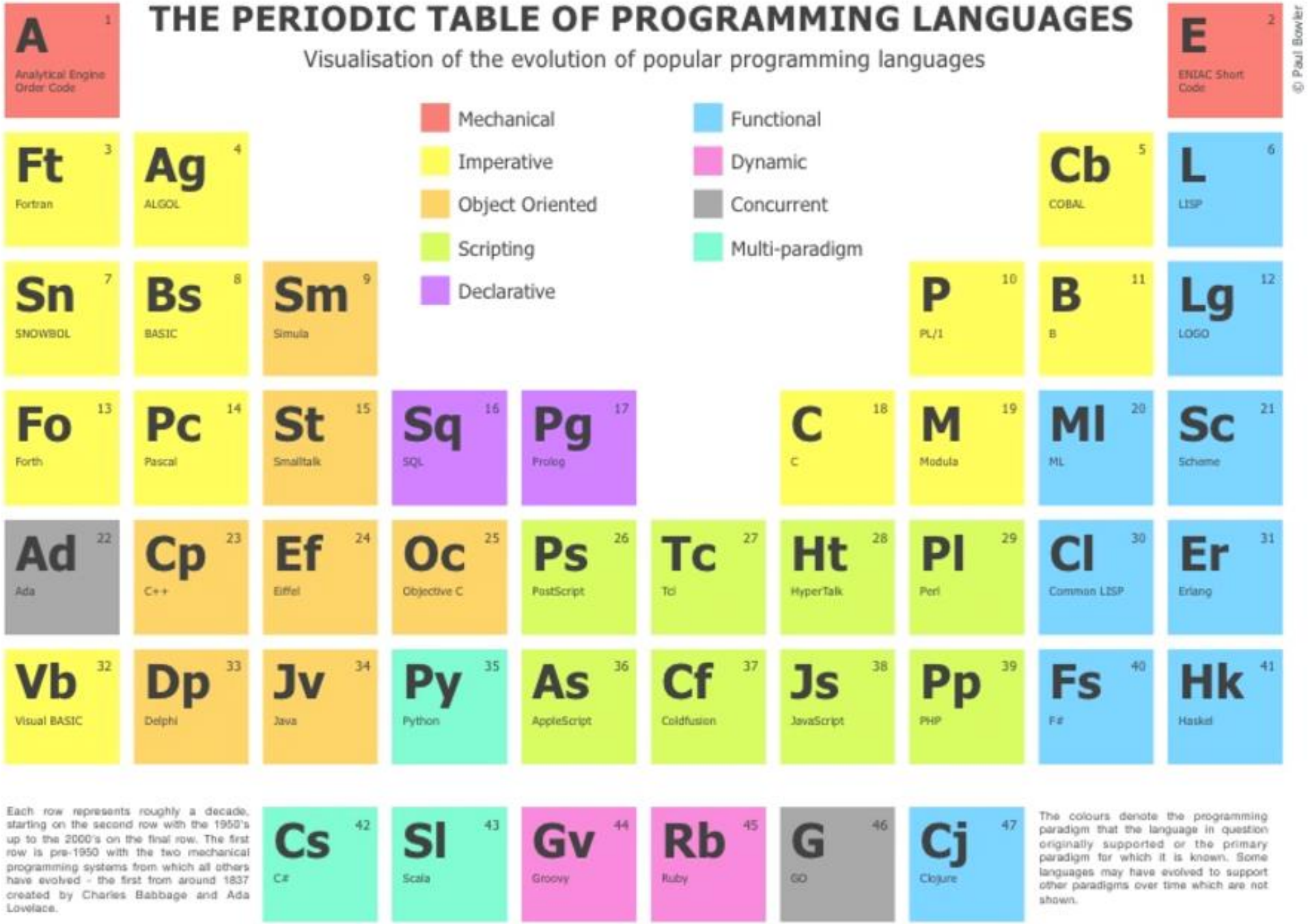
Embedded System

The constraints in the real world of mission-critical embedded systems:

1. Speed
2. Response time (latency)
3. Limited Memory
4. Limited Power

THE PERIODIC TABLE OF PROGRAMMING LANGUAGES

Visualisation of the evolution of popular programming languages



Each row represents roughly a decade, starting on the second row with the 1950's up to the 2000's on the final row. The first row is pre-1950 with the two mechanical programming systems from which all others have evolved - the first from around 1837 created by Charles Babbage and Ada Lovelace.

The colours denote the programming paradigm that the language in question originally supported or the primary paradigm for which it is known. Some languages may have evolved to support other paradigms over time which are not shown.

http://www.slideshare.net/paulbowler/periodic-table-of-programming-languages-13738602

How Coding Work : Example

- Basic Mathematical Problems
- Adding two numbers, 4+3 = 7
- Convert to Equations, a+b = Answer
- Build an algorithm, A little girl asking you how to add two numbers :
    1. ...
    2. ...
    3. ...
    4. ...
- Convert into Program

How Coding Work : Example (cont.)

- Python

```
a = input("Enter first numbers : ")
b = input("Enter second numbers : ")
c = int(a)+int(b)
print("Final result is = " + str(c))
```

- C

```
void main()
{
  int a,b,c;
  printf("Enter two numbers");
  scanf("%d %d",&a,&b);
  c = a+b;
  printf("Final result is = %d",c);
}
```

Convert Python to C / C++

- Add C code into Python
  - Cython
- Convert Python to C/C++ Using
  - Py2c
  - Nutika
  - Shed Skin
- Manual conversion
  - Python to C++ Guide http://personal.denison.edu/~krone/cs173/files/PythontoC++.pdf

Learning to Code

- Increasing my touch-typing speed at typingclub.com.
- Watching a video about some compsci topic, such as those brilliant ones by Computerphile.
- Coding and Coding

Drawing with Turtle in Python

- “Turtle” is a python feature like a drawing board.

https://www.pythonsandbox.com/turtle

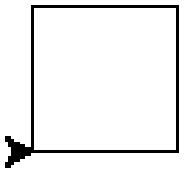
- You can use functions like turtle.forward(…) and turtle.left(…) to move the turtle around.

```
import turtle

t = turtle.Turtle()
t.speed(5) # 1:slowest, 3:slow, 5:normal, 10:fast, 0:fastest
t.forward(100)
t.left(30)
```

Drawing with Turtle in Python

- Draw a square as in the following picture:

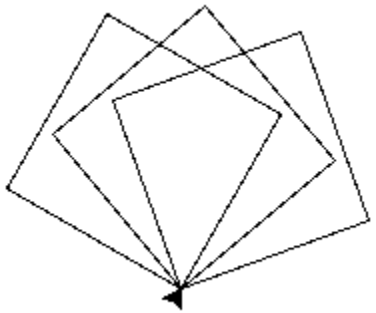


```
import turtle

t = turtle.Turtle()
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
```

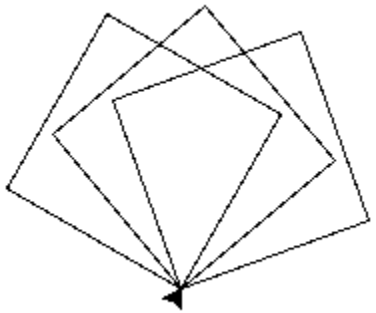
Drawing with Turtle in Python

- Now, draw a tilted square. You can experiment with the angles between the individual squares.



Drawing with Turtle in Python

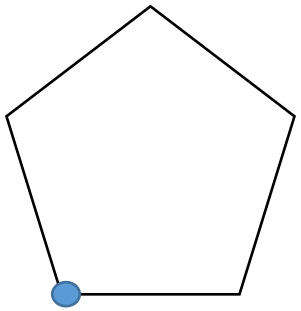
- Draw a tilted square by using function



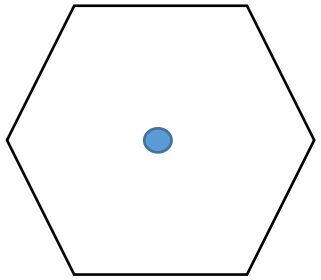
Task 1-1

- Write a function for drawing a n-polygon with equal side length s at position (x, y)

```
polygon(t, n, s, x, y)
```



Type 1



Type 2