

Introduction

COMP360 Programming Languages

This is *real* Computer Science

- This course combines theory, programming, architecture and other aspects of Computer Science
- We will look “under the hood” and learn how programming languages work
- We will make our own computer language

The Way We Communicate

- **Language** is the ability to acquire and use complex systems of communication, particularly the human ability to do so. [Wikipedia]
- Languages can be oral, written or graphical
- Programming languages are the way we communicate an algorithm
 - to a computer
 - to other people



ACM/IEEE Curriculum Guidelines

- Programming languages are the medium through which programmers precisely describe concepts, formulate algorithms, and reason about solutions
- In the course of a career, a computer scientist will work with many different languages, separately or together
- Software developers must understand the programming models underlying different languages and make informed design choices in languages supporting multiple complementary approaches

ACM/IEEE Curriculum Guidelines

- Computer scientists will often need to learn new languages and programming constructs, and must understand the principles underlying how programming language features are defined, composed, and implemented
- The effective use of programming languages, and appreciation of their limitations, also requires a basic knowledge of programming language translation and static program analysis, as well as run-time components such as memory management

Many Languages

- There are over 5,000 human languages
- There are hundreds of programming languages
- Many programming languages have come and gone over time

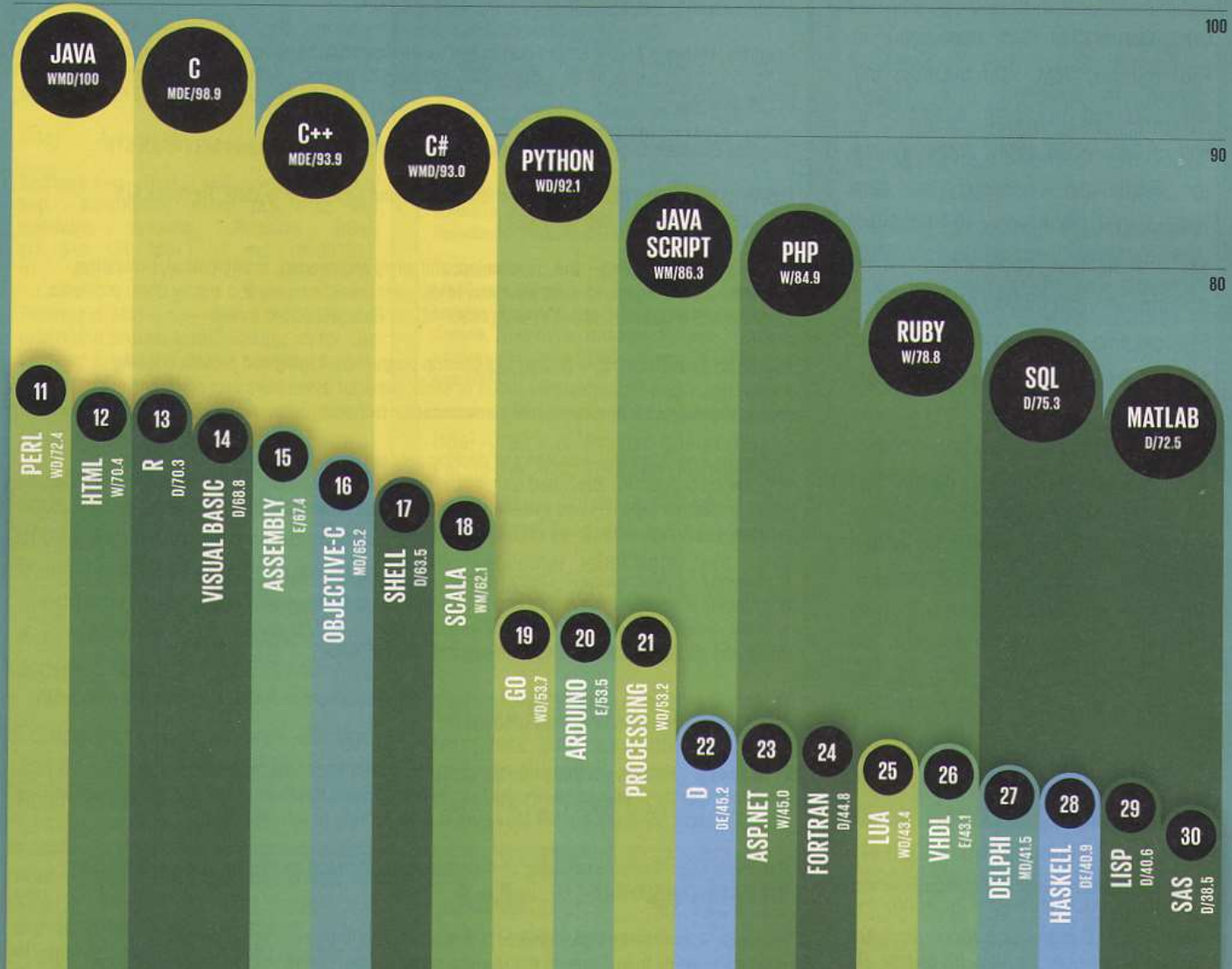
THE TOP 10 PROGRAMMING LANGUAGES

SPECTRUM'S 2014 RANKING

Working with computational journalist Nick Diakopoulos, IEEE Spectrum has weighted and combined 12 metrics from 10 sources (including IEEE Xplore, Google, and GitHub) to rank the most popular programming languages. If you don't agree with our weighting, want to see more languages, or are interested in what's dominant in a specific subsector, such as mobile, go to our online interactive version. There you can adjust the weight of each metric and create your own custom ranking. —STEPHEN CASS

KEY | W-Web | M-Mobile
D-Desktop and Enterprise
E-Embedded

Scores are normalized so that the top-ranked language's score is set to 100.



IEEE 2016 Programming Language Rankings

Language Types



Web



Mobile



Enterprise



Embedded

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9
11. Swift		70.1
12. Arduino		69.9
13. Assembly		68.6
14. Matlab		68.5
15. Scala		66.9
16. HTML		65.6
17. Perl		58.5
18. Visual Basic		56.8
19. Shell		54.2
20. Objective-C		53.4

Why so many programming languages?

- Experience using the existing languages indicates how they can be improved
- Not every language is for every purpose

What makes a good programming language?

- What are the properties of a good programming language?
- What makes it easier to program in one language than another?
- How do you know a good programming language when you see one?

What makes a good programming language?

- Form into groups of 3 -4 students and make a list of the properties of a good programming language

Some attributes of a good programming language

- **Clarity** – You can write the algorithm as you envision it in the application domain
- **Readability** – Someone reading a program will understand what it does
- **Intuitiveness** – It works the way you expect
- **Safety** – The language minimizes the chance to make subtle errors
- **Large library** – Many functions are available
- **Support for abstraction** – Allows encapsulating concepts

What Makes a Language Successful?

- Ease of use for a novice
- Ease of implementation
- Standardization
- Open Source
- Excellent compilers
- Economics, patronage and inertia

Read before class

Read before **Friday**

- https://en.wikipedia.org/wiki/History_of_programming_languages

Read before **Wednesday**

- Textbook chapter 1