

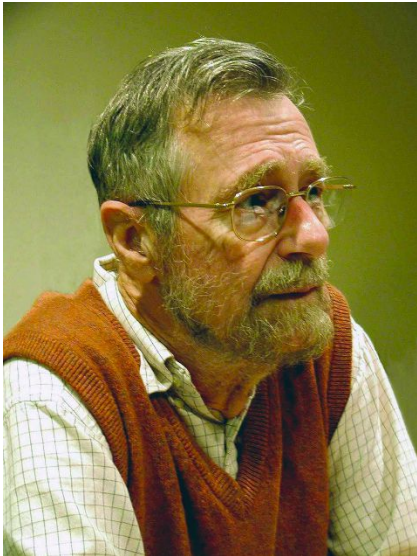
# **Object-Oriented Programming**

**CSCI-GA 3033-021**

Instructor: Thomas Wies

Spring 2018

Class 1 - Introduction



*Object-oriented programming is an exceptionally bad idea which could only have originated in California.*

Edsger Dijkstra

# Object-Oriented Programming (OOP)

*Object-oriented programming is claimed to promote **greater flexibility and maintainability** in programming, and is widely popular in **large-scale software engineering**.*

Wikipedia



# The Goal of this Course

Learn how to build and evolve large-scale programs using object-oriented programming

- **OOP Design:** How do we think in objects?
  - design patterns, object-functional design, objects and concurrency, ...
- **OOP Primitives:** How do we express object orientation?
  - objects, classes, method dispatch, inheritance, subtyping, generics, ...
- **OOP Implementation:** How do we realize OO primitives?
  - virtual method dispatch with vtables, static overloading resolution, type erasure, ...

# How Do We Achieve This Goal?

- In-class lectures and discussions
  - lectures to introduce topics and techniques
- Individual homework assignments
  - gives a structured introduction to concepts
  - one week for completion
- Mini projects:
  - Teams of 2 students
  - two or more weeks for completion

# Used Language: Scala (v. 2.12)

- Scala fuses the **object-oriented** and **functional** programming paradigms
- This is a general trend in languages that are used in industry (see Java 8, C# 6.0, Swift, ...)
- Scala goes furthest in this trend
- It has perhaps the most avant-garde and advanced features of all OOP languages
- Many of these features are extremely useful for writing maintainable and efficient large-scale software
- We will occasionally look at other OOP languages when we discuss design choices in the implementation of OOP features.

# Why Scala?



UBS

twitter

Linked ™

NETFLIX

APACHE  
*Spark*™

  
airbnb

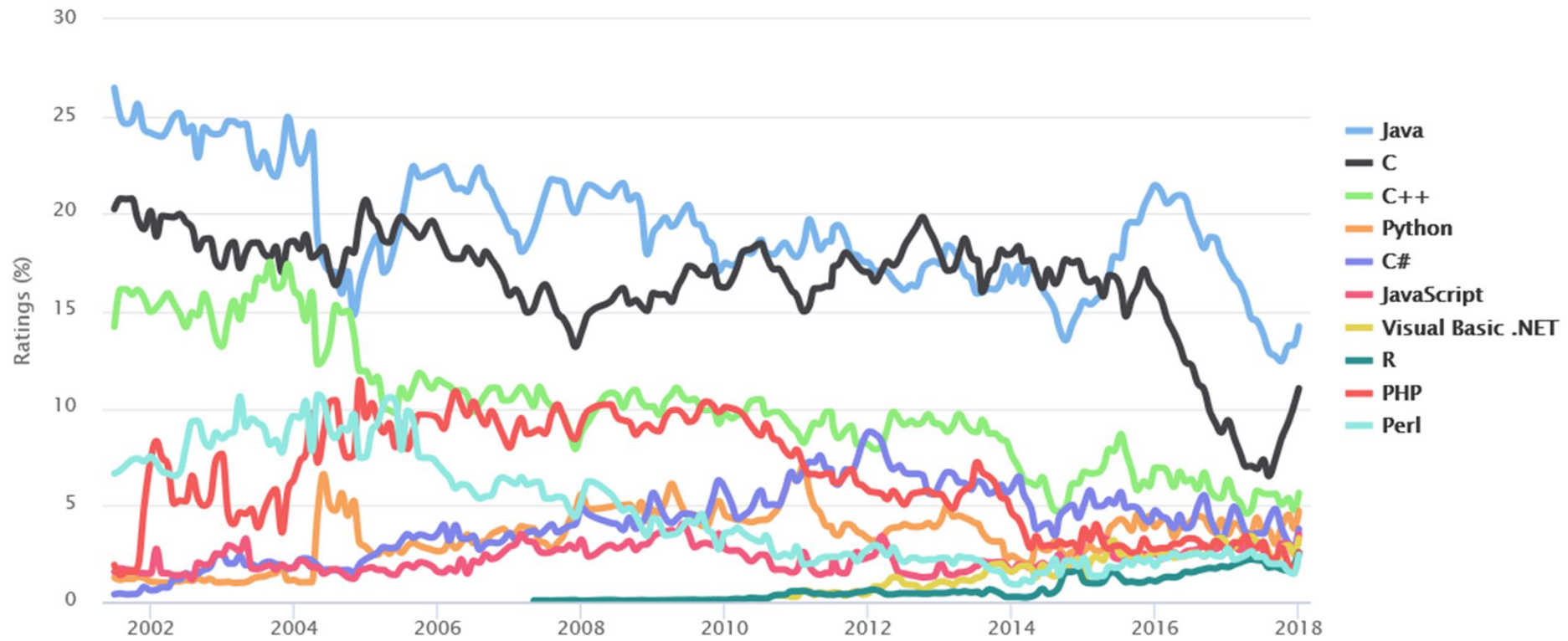
coursera

Novell®

# Programming Language Popularity

TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)





# Operational Details

# Important Dates

- Class: Wed 7:10 - 9pm in WWH 201
- Office hour: Wed 4:00 - 5:00pm in 60FA 403
- Final Exam: Wed, **May 2**
  - last regular class period (i.e., **not May 9**)!
  - no midterm exam

# Textbook

- Programming in Scala, Bill Venners, Lex Spoon, and Martin Odersky, 3<sup>rd</sup> ed., 2017
  - You can find older editions on the web, which will also suffice for most of our purposes
- Rather than making you buy more books I will rely on free online resources where I can

# Online Resources

- **Piazza** - Online discussion and announcements
- **NYU Classes** - Grade posting
- **Github** – Homework assignments, project, and class notes and source code
- **Website**
  - Lists reading assignments, class notes
  - Provides links to useful material

# Grading

- 50% for homework assignments and projects
- 50% for final exam

# Submission Policies

- Grading criteria for projects and homework assignments will be published.
- Solutions must be submitted before the announced date and time deadline for full credit.
- For every 24 hours late you lose 10%
- Late solutions will not be accepted after the late deadline. (usually one week).
- If you turn in a solution that does not compile, it will not be accepted. You can resubmit according to the above rules.

# Rules & Resources

- You must do all homework assignments on your own, without any collaboration!
- Pair programming on the mini projects is strongly encouraged.
- Keep all solutions to projects/homeworks within the private repositories of the Github organization for the course. Do not make them publicly available!
- If you need help, your first stop is Piazza. If you have a question, then almost certainly someone else has it, too.
- Otherwise, see me during the office hours (or make an appointment if the time slot does not work).

# Toolchain

- Linux or OS X.
  - I will give instructions and support for Ubuntu and OS X.
  - I will provide instructions on installing a VM for Ubuntu on Windows.
- IntelliJ + Scala Plugin
  - While using an IDE is not mandatory, I recommend to use these as it greatly improves your programming experience.
  - Full version is available for free under a student license.
  - Alternatives: Scala IDE (Eclipse), Emacs, Vim, ...
- Sbt, ScalaTest, Git, ...
  - Real software engineering tools!
  - Your first homework will be a detailed guide on installing most of these tools.
  - You will need them!!
- Homework 1 will deal with setting up the toolchain.



# What you should do immediately

- You should have received an invite to the Piazza course website.
- If you did not receive an invite, please contact me immediately.
- Complete the questionnaire that I posted on Piazza. Otherwise, you won't have access to the course materials on Github.