C++: Syllabus & Introduction

Outline

- Course Introduction
- C++ Introduction

Who I am

- Instructor: Keith Perkins
- Office: Luter 207
- Office Hours: 11:50 - 1:00 M, W
- 10:45 – 12:00 T,Th
- 1:00 – 2:00 F
- Office Phone: 594-8425
- Email: keith.perkins@cnu.edu 

Notes, Lectures, Assignments, Videos …

- Assignments on Scholar
- Everything else on webpage https://cnuclasses.github.io/CPSC327/
- Note in particular;
  - The Lectures/Readings section
    - You are responsible for everything here
  - The Examples section
    - Understand these

Assignments

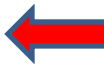
- Read All week 1 readings
- Please install and set up Eclipse CDT
- Please go to projects section of website
  - Complete Project 1 by due date
  - Mind the submission format (you will use this to import my projects)

Syllabus: Prerequisites

- CPSC 250 or equivalent
- Textbook – Any C++ text
- Suggestions:
  - Absolute C++ 5th edition, Walter Savitch
  - C++ Programming Language, Stroustrup
- References to make you a better programmer
  - Effective C++, Scott Meyers
  - More Effective C++ , Scott Meyers
  - Effective STL, Scott Meyers
  - Effective Modern C++, Scott Meyers

Syllabus: Major Topics
(Subject to change)

- Week 1 — Intro, Market share, Compilation, GIT, Linux introduction
- Week 2, 3 — Some C++ Syntax, Standard Library, Strings
- Week 4 — Headers, functions, Streams, Structs, Enums
- Week 5 — Standard Library iterators and Lists, Preprocessor directives
- Week 6, 7 — Pointers, References, Memory
- Week 8 — Classes, operators, memory management using RAII
- Week 9 — Exceptions
- Week 10, 11 — Inheritance, operator overloading, virtual heiarchys
- Week 12,13 — Concurrency and Threading
- Week 14 — Registers, Memory, profiling



Syllabus : Evaluation

- 1 Midterm Test
- 1 Final
- Numerous projects
- See Syllabus for details

- This will be a rigorous course. Please start projects early.

Syllabus: Assignments

- Project 1 - 50 points - Install and demo IDE
- Project 2 - 100 points - File I/O
- Project 3 - 100 points - STD library containers and Sorting
  Project 4 - 200 points - Using libraries and parsing strings
- Project 5 - 250 points – Polymorphism

Development Environment

- Could use vim, g++, gdb, valgrind, tmux for a command line only dev environment
- Or an IDE, Lots to choose from, Codeblocks, Netbeans,Ms Visual Studio, Eclipse CDT…Clion
- We will use Eclipse CDT

What you will learn

- Standard C++ - to a level of proficiency so you can function professionally, you will not be an expert.
- Some of the C++ syntax
- Coding suggestions and Guidelines to make you a better programmer.
- how to use an IDE, how to use libraries, how to approach and solve programming problems

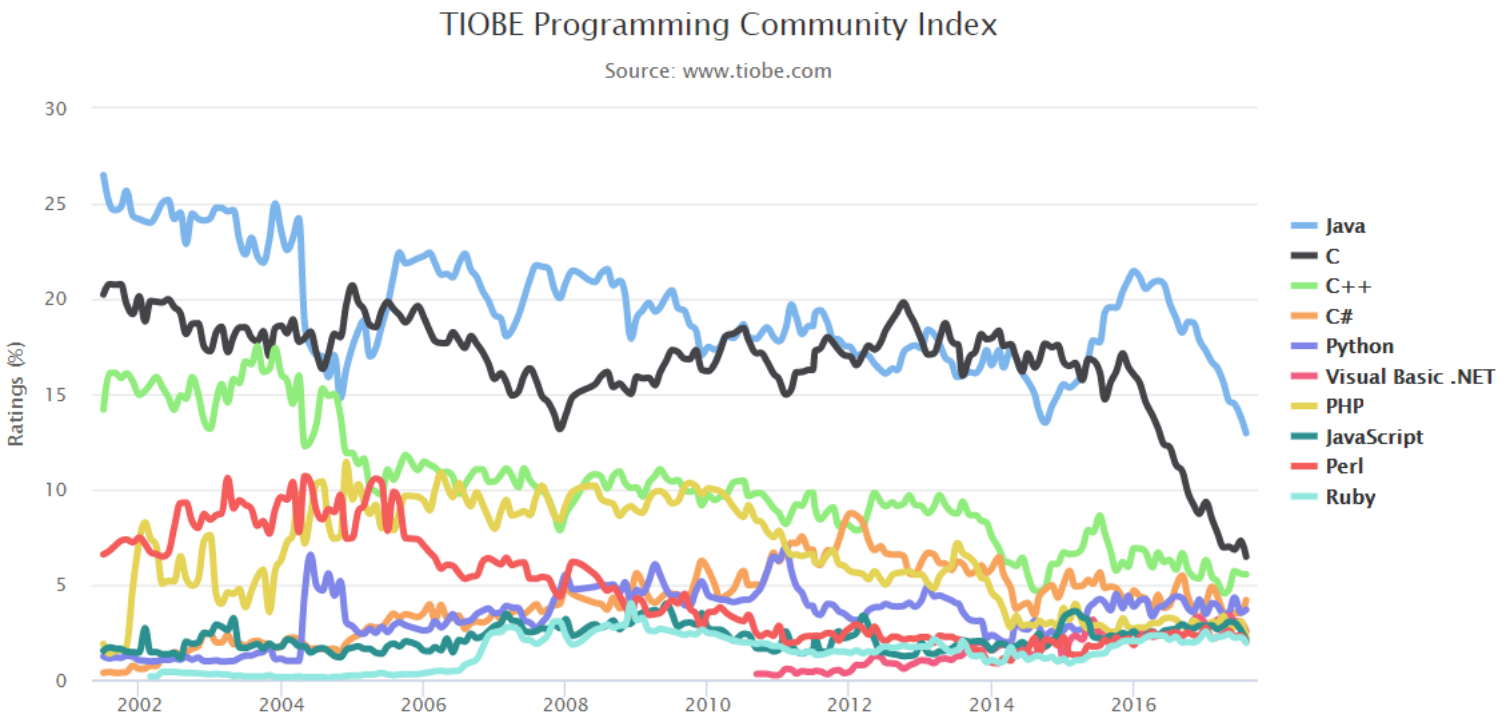
What you will NOT learn

- User Interface (UI), networking– UI is platform dependent, networking is too advanced for intro class (and is MUCH harder in C++ than Java)

Outline

- Course Introduction
- C++ Introduction

C++ Usage

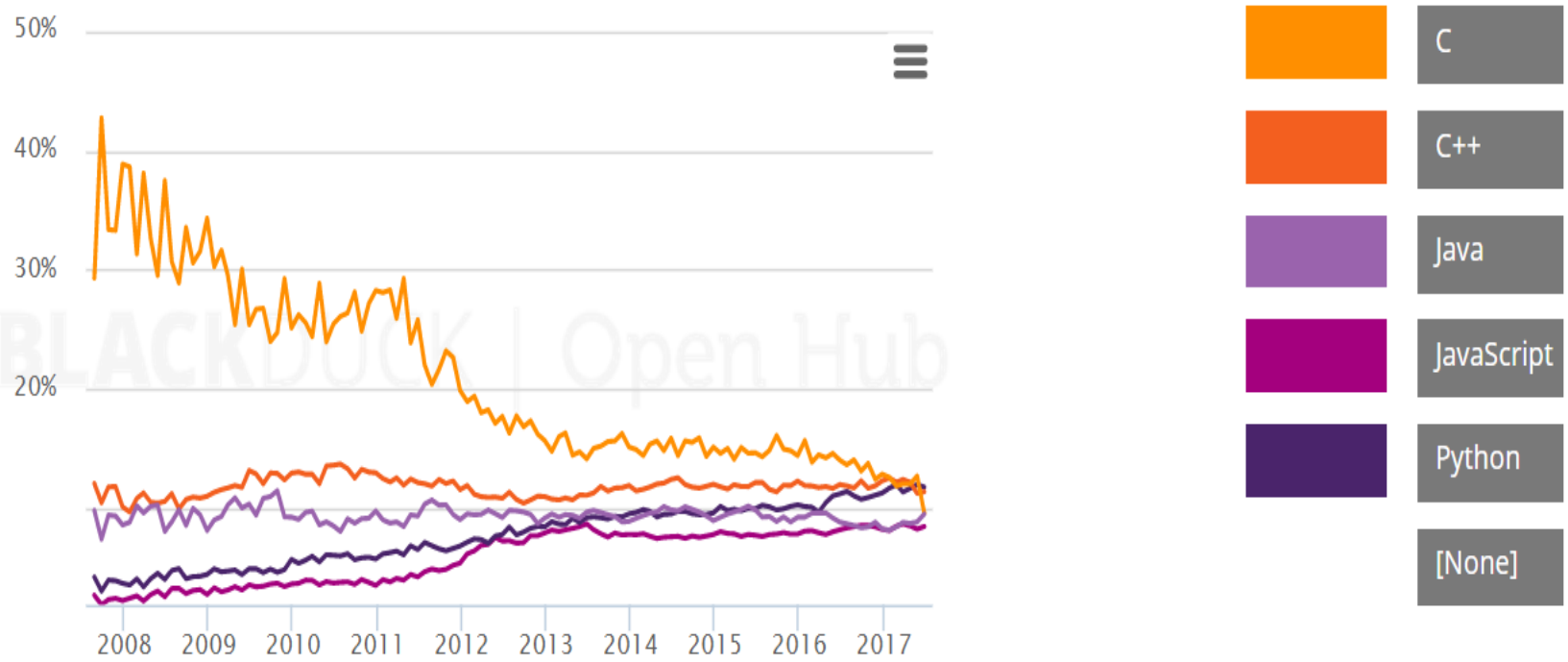


see http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

C++ Usage

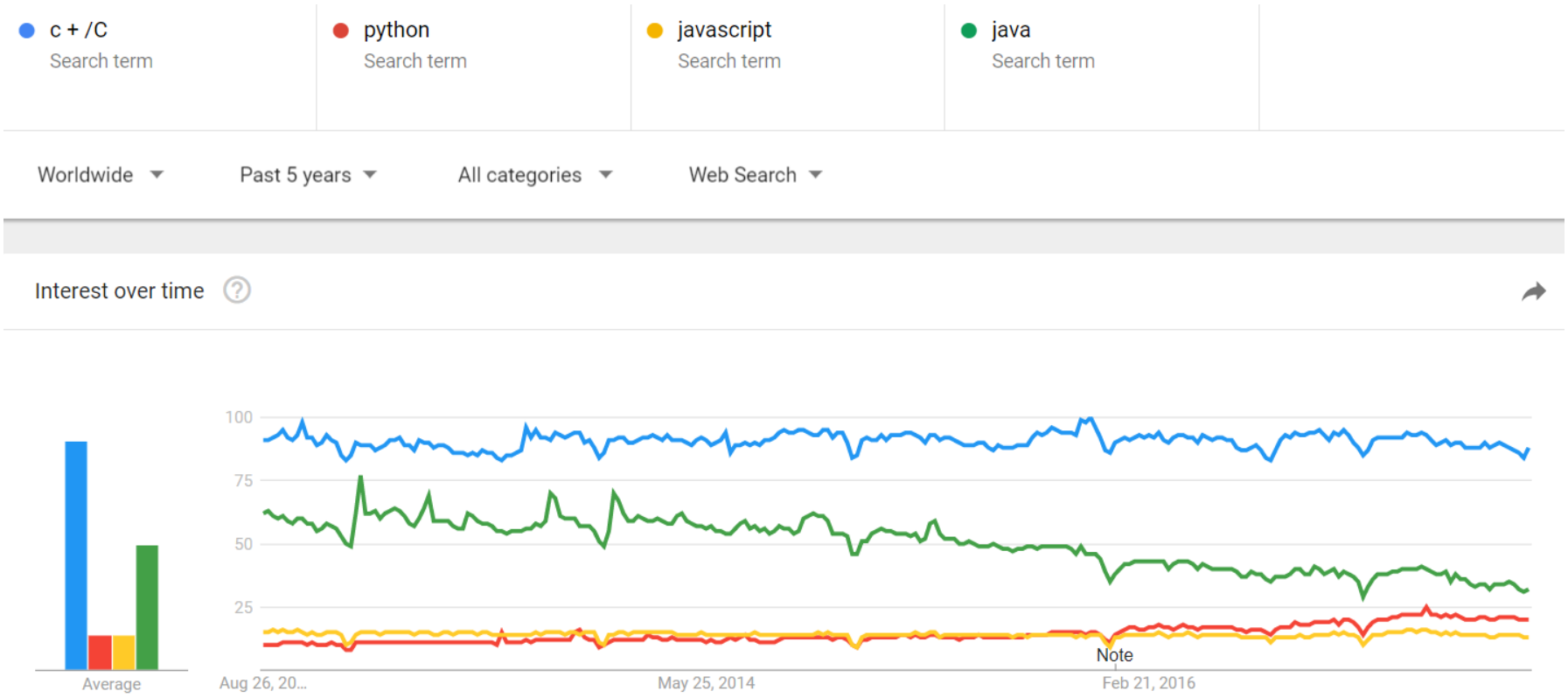
Monthly Commits (Percent of Total)

The lines show the count of monthly commits made by source code developers. Commits including multiple languages are counted once for each language.



See https://www.openhub.net/languages/compare

C++ Usage



See https://www.google.com/trends/explore?date=all&q=c%2B%2B%2FC,python,javascript

C++ … Why?

- Fast
- You have absolute control over everything
- Elegant when done well
- Only choice for some situations
  - High speed trading
  - Google search
  - Embedded systems
  - Real Time Processing
- Low level control

C++ … Why not?

- Harder to code than languages that run on a VM (Java, C#)
- No garbage collection, pointers can be (and usually are) a problem
- Must be compiled to target platform, no portable bytecode
- My experience – My Java apps are up and running much faster than my C++ apps.

C++ … Where is it used?

- Device driver development
- Video Games
- Advanced engines (audio, image processing, etc)
- Telecom
- Embedded software
- Financial - low latency market data feeds
- Google
- Real time video processing

I know Java why bother?

- Speed
- Software now targets distributed applications
  - Rich user interfaces
  - Cloud storage
  - Mobile Applications
  - Big Data
- Today, applications require expertise in multiple languages

But… I don’t know most of that stuff

- Don’t worry, you aren’t expected to.
- You learn on the job (while getting paid)