CS – 211
Programming I for Engineers

Instructor: Tom Bartenstein

Course Web Page:
http://www.cs.binghamton.edu/~tbarten1/CS211_Fall_2015/

Catalog Description

- Introduction to computer programming with engineering applications.
- Programming in the procedural language C,
    - control structures,
    - functions,
    - arrays and pointers.
- Introduction to abstract data types and object-oriented programming using C++.

Textbooks

- Kochan,
  - “Programming in C, Fourth Edition”, Addison Wesley, 2015
- Brian Kernighan and Dennis Ritchie,
  - “The C Programming Language, Second Edition”, Prentice Hall, 1988

Teaching Staff

- Prof: Tom Bartenstein (tbartens@binghamton.edu)
  - Office Hours: MWF 10:00am – 12:00 Noon @ EB P-14
- Course Assistants
  - Steven Popovitch
  - Yurly Boot

Course Mechanics

- Lectures : Mon/Wed/Fri
  - Attendance Expected!
  - Added incentive: Pop Quizzes (3 or 4 over the semester)
- Labs: 5 sections over Tue/Wed/Thur
  - Attendance Expected
  - Practice coding
  - Complement to Lectures (I will know if you skip!)
- Homework
  - Reading
  - Practice Problems – Self Graded
- Projects
  - 3 or 4 Larger Coding projects

Grading

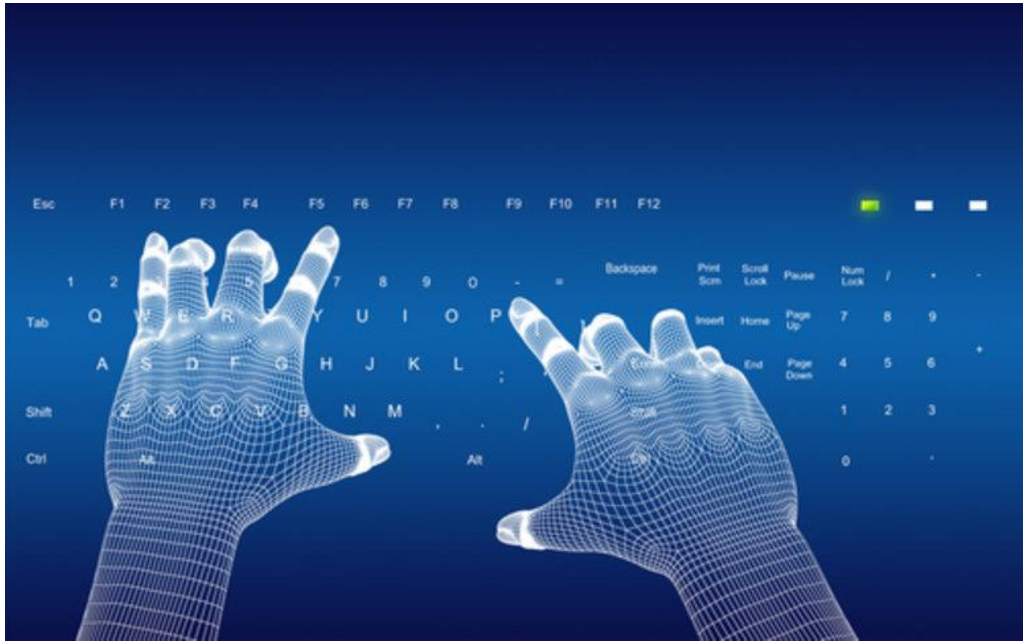
Quizzes, Attendance, Participation	15%
Labs	15%
Projects	30%
Tests	20%
Final Exam	20%

CS-211 Course Goals

- Goal: Learn how to write application software
  - What is “application software?”
  - Why would I want to write [good] application software?
  - How do we learn to write good software?
- Goal: Teach yourself a new(?) computer language
  - What programming languages do you know? Python? Java? C? x86 Assembly? Perl? Ruby? Others?

What is Programming?

- Algorithm
- Language
- Machine Language
- Compilation
- Interpretation
- Abstraction vs. Speed
- Generality vs. Efficieny
- Engineering vs. Art vs. Science

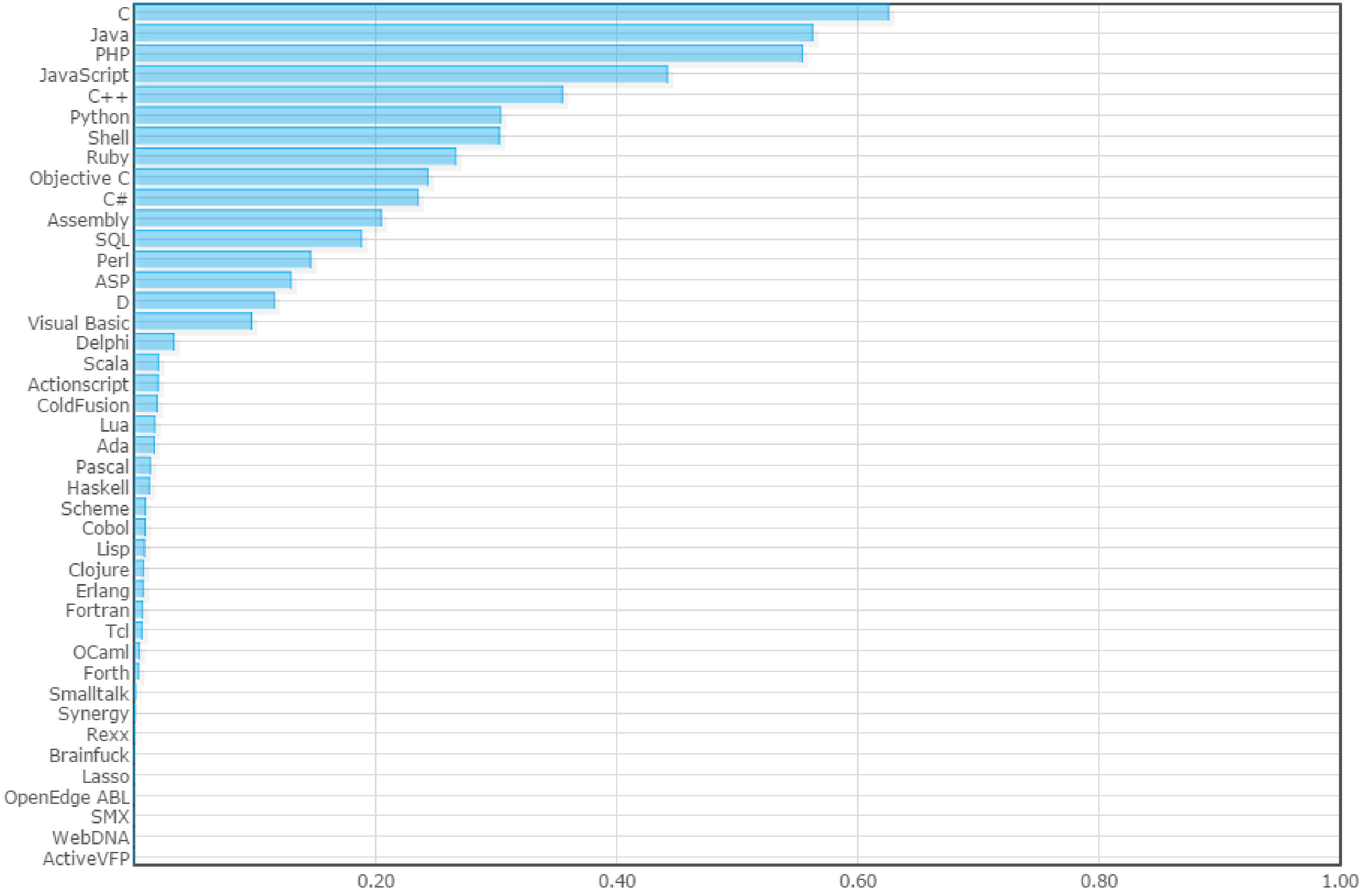


How to learn a Language

- You can’t teach a language!
- You can teach syntax , grammar, and vocabulary
- You cannot teach style/voice
- You cannot learn to program without practice!
  - You can’t read without practice
  - You can’t write without practice

Why C?

- Popularity
- Efficiency
- Simplicity



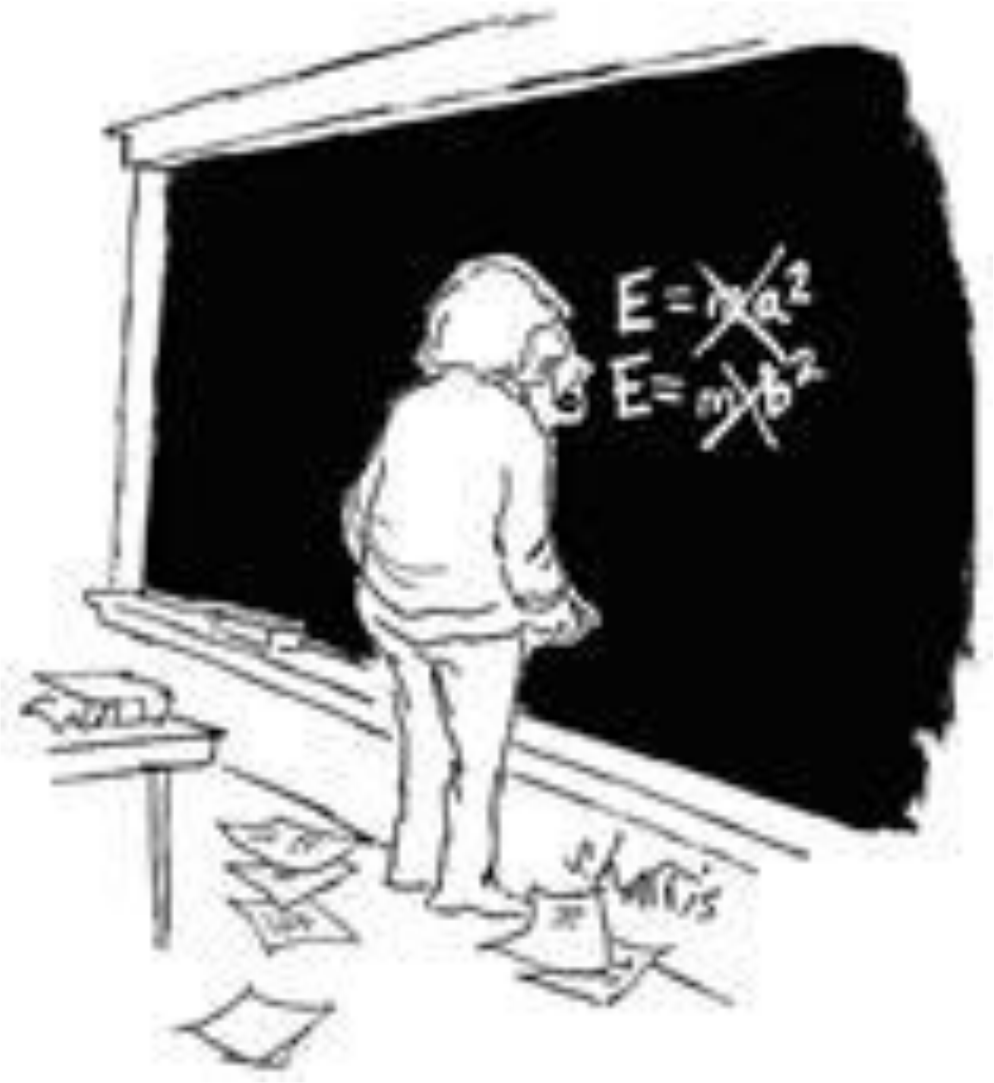
Origins of C

- MULTICs operating system
  - MIT / General Electric / Bell Labs Collaboration, 1964 – 1970
  - Design an operating system to run GE-645 Main Frame
  - Programmed in Assembler and PL/I
- BCPL (Basic Combined Programming Language)
  - Martin Richards (University of Cambridge) 1966
  - Used to write Compilers
  - First use of {}, // , First "Hello World"
  - No Types (interpretation based on operator)

Origins of C

- UNIX (1969-1970’s) Operating System
  - Bell Labs PDP-7, PDP-11
- “B” Language (BCPL “Lite”)
  - Ken Thompson and Dennis Ritchie
  - Still no types

How C got it's name….



Origins of C

- B + types + structures = C
- 1978 “The C Programming Language” (Kernighan and Ritchie) K&R
- 1983 – ANSI Standard
- 1988 – POSIX standard (libraries)
- 1990 – ISO “C89”
- 1999 – ISO “C99”
- 2011 – ISO “C11”