Mathematical Software Programming (02635)

Lecture 1 — September 6, 2018

Instructor: Martin S. Andersen

Fall 2018

Practical information

Format

- 5 ECTS (1 ECTS $\sim$ 28 hours on average)
- Short lectures, focus on exercises (303A-A49)
- Weekly reading assignments (see *Calendar* on Inside)
- Two written hand-in assignments (more info later)
- Final exam (written exam, Dec. 11, 2018)

Instructors

- Martin S. Andersen (`mskan`), DTU Compute
- Bernd Dammann (`beda`), DTU Compute/DCC

Teaching assistants

- Mathias Sorgenfri Lorenz (`s134597`)
- David Frich Hansen (`s144242`)

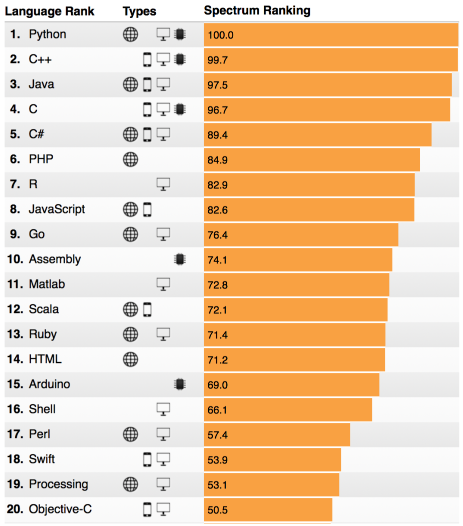
Learning objectives

- Evaluate discrete and continuous mathematical expressions.
- Describe and use data structures such as arrays, linked lists, stacks, and queues.
- Choose appropriate data types and data structures for a given problem.
- Compare iterative and recursive solutions for simple problems.
- Analyze the run-time behavior and the time and space complexity of simple programs.
- Call external (third party) programs and libraries.
- Design, implement, and document a program that solves a mathematical problem.
- Debug and test mathematical software.
- Describe and use basic object-oriented programming concepts such as classes and objects.

Why C?

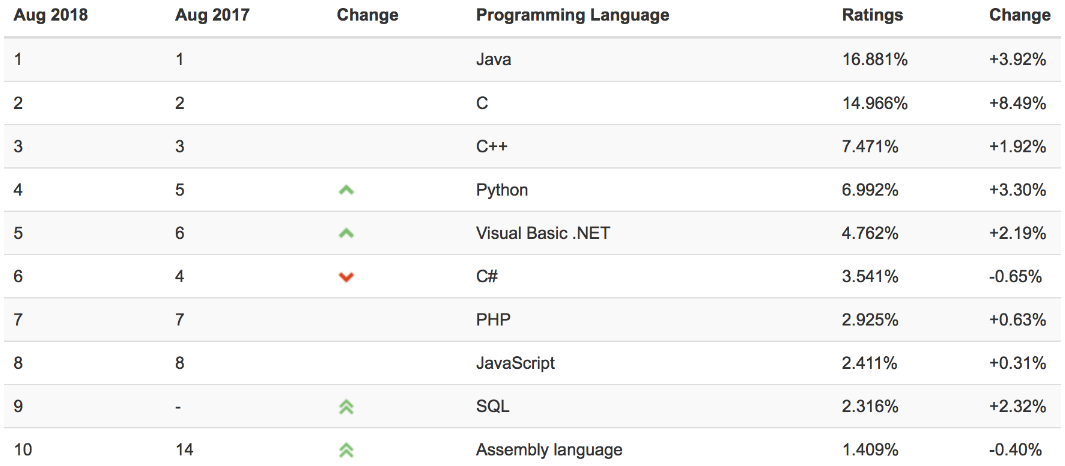
- Widely used and mature programming language (developed in the early 1970s)
- Industry standard (ANSI C (C89) / ISO C (C90), C95, C99, C11, C18)
- Many newer programming languages are syntactically similar to C (e.g., C++, C#, Objective C, Java, PHP, Go, . . .)
- Cross-platform support
- Low-level control (direct access to low level hardware/APIs)
- Low overhead (high performance)
- Statically typed language
- Understanding of memory management (no “magic” under the hood)
- Embedded systems (IoT)
- C *powers* the world (OS kernels, Python, MATLAB, . . .)

IEEE: The Top Programming Languages 2018



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥 ▣	100.0
2. C++	📱 🖥 ▣	99.7
3. Java	🌐 📱 🖥	97.5
4. C	📱 🖥 ▣	96.7
5. C#	🌐 📱 🖥	89.4
6. PHP	🌐	84.9
7. R	🖥	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥	76.4
10. Assembly	▣	74.1
11. Matlab	🖥	72.8
12. Scala	🌐 📱	72.1
13. Ruby	🌐 🖥	71.4
14. HTML	🌐	71.2
15. Arduino	▣	69.0
16. Shell	🖥	66.1
17. Perl	🌐 🖥	57.4
18. Swift	📱 🖥	53.9
19. Processing	🌐 🖥	53.1
20. Objective-C	📱 🖥	50.5

TIOBE Index August 2018



Aug 2018	Aug 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.881%	+3.92%
2	2		C	14.966%	+8.49%
3	3		C++	7.471%	+1.92%
4	5	▲	Python	6.992%	+3.30%
5	6	▲	Visual Basic .NET	4.762%	+2.19%
6	4	▼	C#	3.541%	-0.65%
7	7		PHP	2.925%	+0.63%
8	8		JavaScript	2.411%	+0.31%
9	-	⏫	SQL	2.316%	+2.32%
10	14	⏫	Assembly language	1.409%	-0.40%

Resources

Textbooks

- S. Oliveira & D. Stewart, "Writing Scientific Software: A Guide to Good Style", 2006
  - ISBN: 9780521675956
- I. Horton, "Beginning C", 5th ed., 2013
  - ISBN: 9781430248811
  - E-book available through DTU Library
  - Source code available for examples

Supplementary resources (optional)

- M. Olsson, C quick syntax reference, 2015
- I. Horton, Beginning C++, 2014
- M. Olsson, C++ quick syntax reference, 2013
- OnlineProgrammingBooks.com
- Big-O Cheat Sheet
- Learn to Solve It: C programming exercises

Help!?

Instructors/teaching assistants

- Be prepared
- Write down questions
- Get feedback

Piazza

- Post your (anonymous) questions on Piazza discussion board
- Learn from and help your peers

Email

- Please use email for personal matters only

Documentation and reference manuals

- GNU C Library
- GNU C Library - function index
- GNU Compiler Collection (GCC) Manual
- Wikipedia: C mathematical functions
- GNU Scientific Library
- Cplusplus.com
- Cprogramming.com
- Boost C++ Library

Compilers

Compiler installation guide available on Inside

- Linux/Unix
  - GCC (Ubuntu/Debian: `sudo apt-get install build-essential`)
  - `clang` (`sudo apt-get install clang`)
- Mac OS X
  - Clang (`xcode-select --install`)
  - GCC (e.g., via Homebrew)
- Windows
  - GCC via MSYS2
  - GCC via Windows Subsystem for Linux (WSL)
  - Visual Studio C++ (no support)

Software

Cross-platform editors & IDEs

- Atom
- Visual Studio Code
- GNU Emacs
- Vim
- Eclipse

Tools

- GNU Make
- GNU debugger
- GNU profiler
- Valgrind profiler

DTU Resources

- gBar DataBar
- DTU Computing Center

Historical Perspective

C89	C99	C11	C18
Intel 80486	Intel Pentium III	Intel Core i7 (1st gen)	Intel Core i9 7980XE
$350	$800	$600	$2000
0.03 GFLOPS	1-2 GFLOPS	80 GFLOPS	950 GFLOPS
Macintosh Portable	PS2	iPhone 4s	iPhone 8
$6,500	$299	$650	$699
No FPU	6 GFLOPS	12 GFLOPS	300 GFLOPS

- GCC version 8 supports C18
- Microsoft Visual Studio 2017
  “*. . . generally compatible with the ISO C99 standard, but not strictly compliant.*” (VS17)

Today's exercises

Available under *File sharing* on Inside

- Part I: install a C compiler and a text editor
- Part II: do the exercises (individually or in small groups)

If you finish early, start preparing for next week!

Compile and run “Hello 02635!” program

Create a plain text file `hello.c` with the following code:

```c
#include <stdio.h>

int main(void) {
    printf("Hello 02635!\n");
    return 0;
}
```

Compile and run your program:

```
$ gcc -Wall -std=c99 hello.c -o hello
$ ./hello
```

Compiling “Hello World” with GNU Make

```
$ make hello
gcc     hello.c   -o hello
```

```
$ make "CFLAGS=-std=c99 -Wall" hello
gcc -std=c99 -Wall    hello.c   -o hello
```

Makefiles

Create a plain text file and call it `Makefile` (no extension!)

```
CFLAGS= -std=c99 -Wall   # Extra flags for the C compiler
LDLIBS=                  # Extra library flags (e.g. -lm)
```

```
$ make hello
gcc -std=c99 -Wall    hello.c   -o hello
```

Using the Atom editor

Installation

- Install Atom (provides two commands: `atom` and `apm`)
- Open Atom and install the gcc-make-run package, or using the `apm` command-line tool:

```
$ apm install gcc-make-run
```

- Compile and run your program with `f6`
- Set compiler/options using `ctrl-f6` or `cmd-f6`

Useful Atom extensions

- linter (flag suspicious code): `linter-gcc`, `linter-clang`
- auto-indentation: `atom-beautify`
- auto-complete (Clang users): `autocomplete-clang`
- highlight current selection: `highlight-selected`
- source code preview: `minimap`