



Object Oriented Programming by C++

C++ Basic (1/2)

Orientation, and Basic Structure of C++ Program

2017. 8.

Sungwon Lee / Professor

Email: drsungwon@khu.ac.kr

Web: <http://mobilelab.khu.ac.kr/>

Modified & taught by Jinwoo Choi
in 2023 Spring

Email: jinwoochoi@khu.ac.kr

Webpage: <https://sites.google.com/site/jchoivision/>

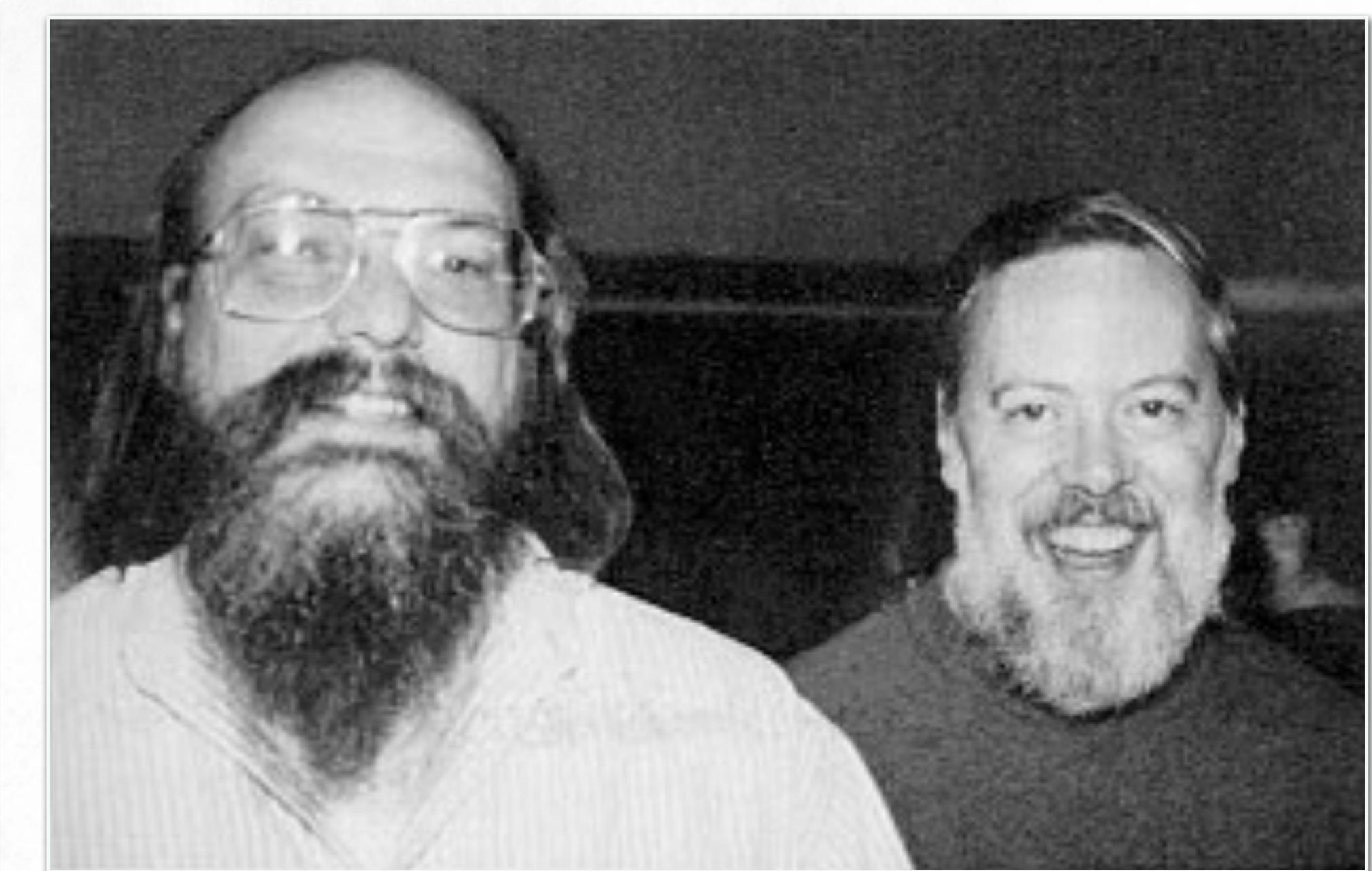
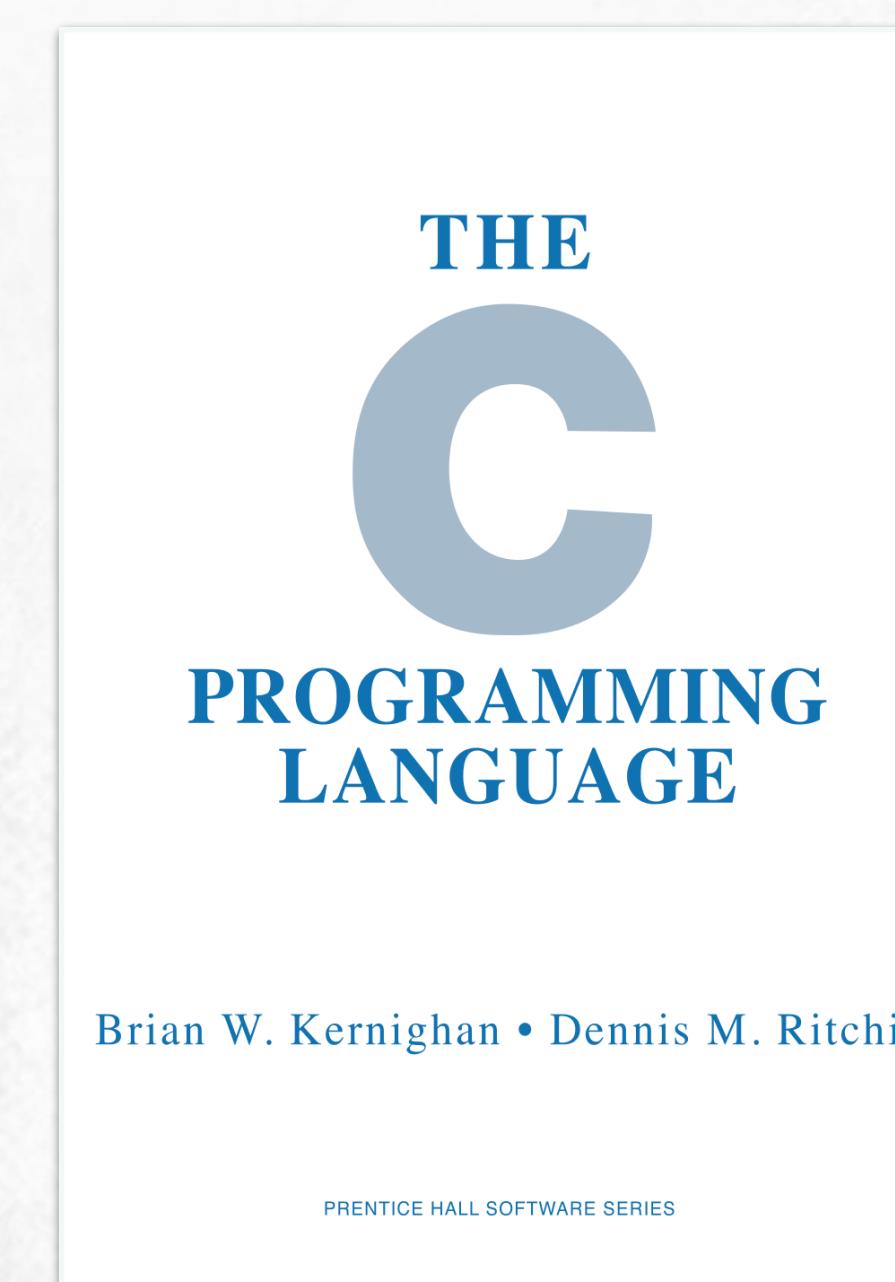
Contents

- C & C++ Language
- Language Popularity
- Software Build Process and Tools
- Open Source Resources for Software Programming
- Example Program - *Empty*
- General Structure of C++ Program
- Software Development Stage

C Language

C was

- Originally developed by Dennis Ritchie between 1969 and 1973 at Bell Labs, and used to re-implement the Unix operating system
- It has since become one of the most widely used programming languages of all time, with C compilers from various vendors available for the majority of existing computer architectures and operating systems
- It has been standardized by the American National Standards Institute (ANSI) since 1989 (see ANSI C) and subsequently by the International Organization for Standardization (ISO)



C Language

C is

- An imperative procedural language
 - It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support
- Despite its low-level capabilities, the language was designed to encourage cross-platform programming
 - A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code
 - The language has become available on a very wide range of platforms, from embedded microcontrollers to supercomputers

C++ Language

C++ was

- In 1979, Bjarne Stroustrup, a Danish computer scientist, began work on "C with Classes", the predecessor to C++
- Initially, Stroustrup's "C with Classes" added features to the C compiler, Cpre, including classes, derived classes, strong typing, inlining and default arguments
- In 1983, "C with Classes" was renamed to "C++"
- In 1985, the first edition of The C++ Programming Language was released, which became the definitive reference for the language, as there was not yet an official standard
- In 1989, C++ 2.0 was released, followed by the updated second edition of The C++ Programming Language in 1991
- As of 2017, C++ remains the third most popular programming language, behind Java and C



C++ Language

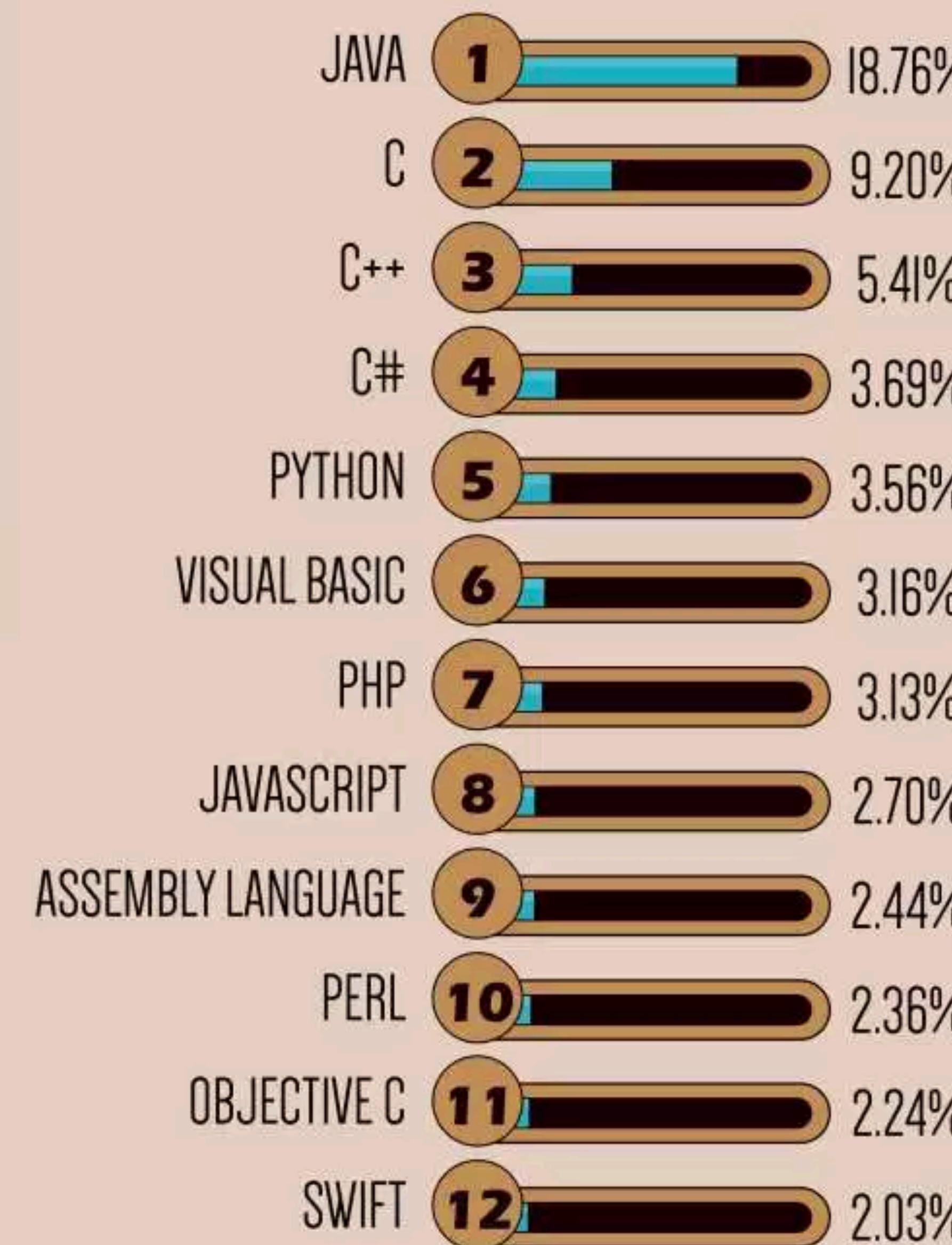
C++ is

- General-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation
- Bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights
- C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes)
- C++ is a compiled language, with implementations of it available on many platforms
- Many vendors provide C++ compilers, including the Free Software Foundation, Microsoft, Intel, and IBM
- Many other programming languages have been influenced by C++, including C#, D, Java, and newer versions of C

Language Popularity

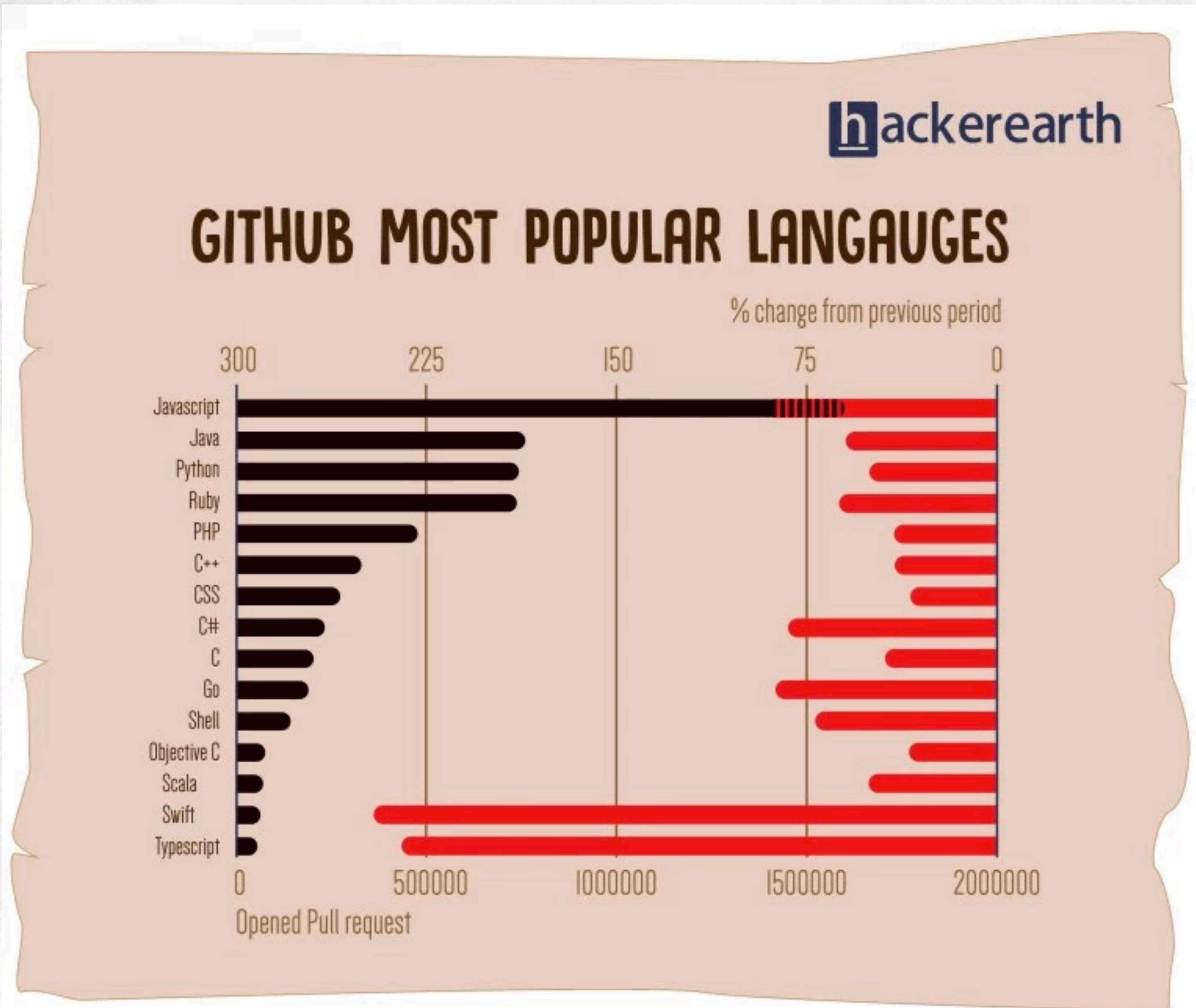
TIOBE Index

TIOBE TOP LANGUAGES AS ON NOV 2016



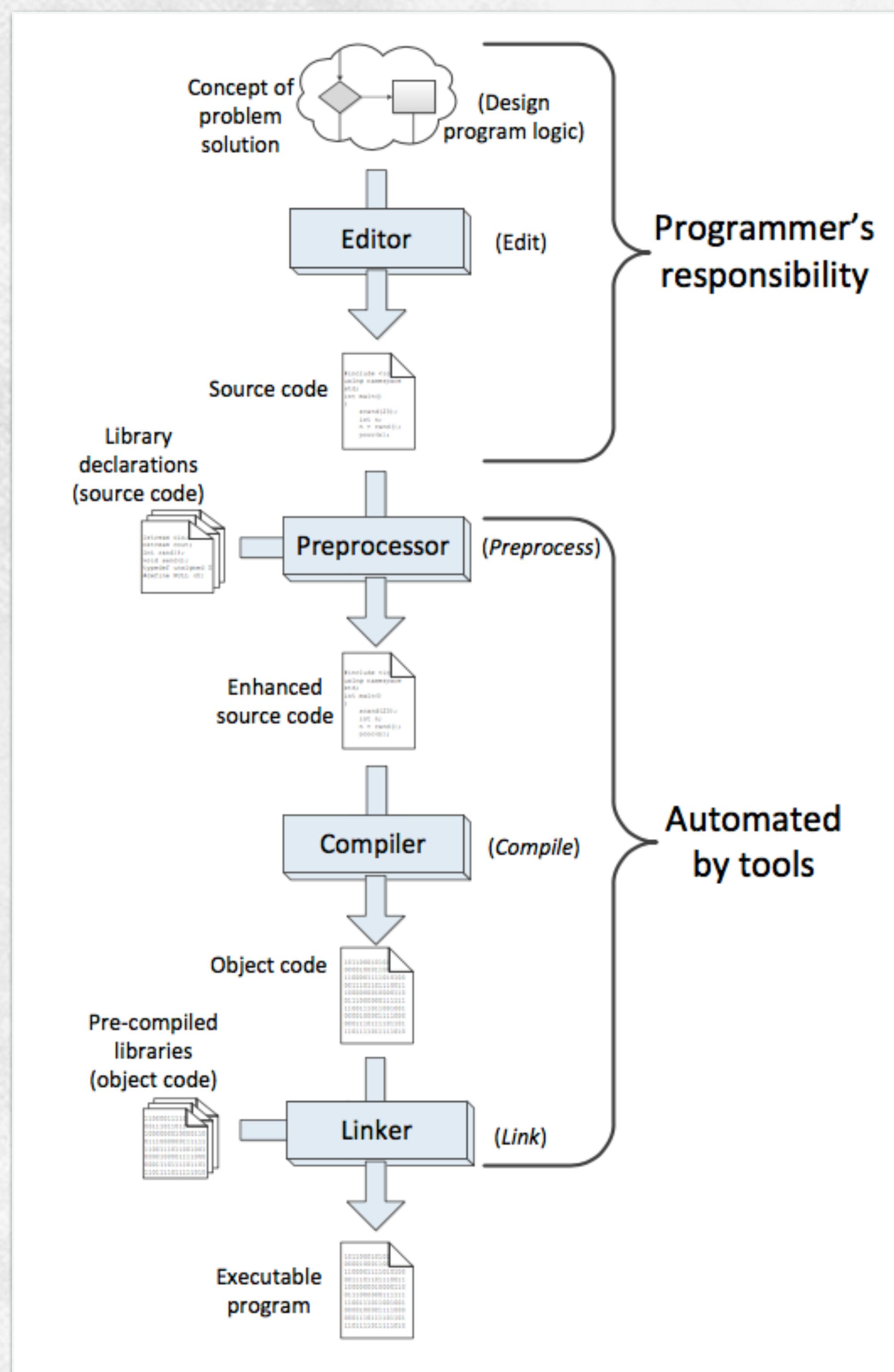
Language Popularity

GitHub



Software Build Process & Tools

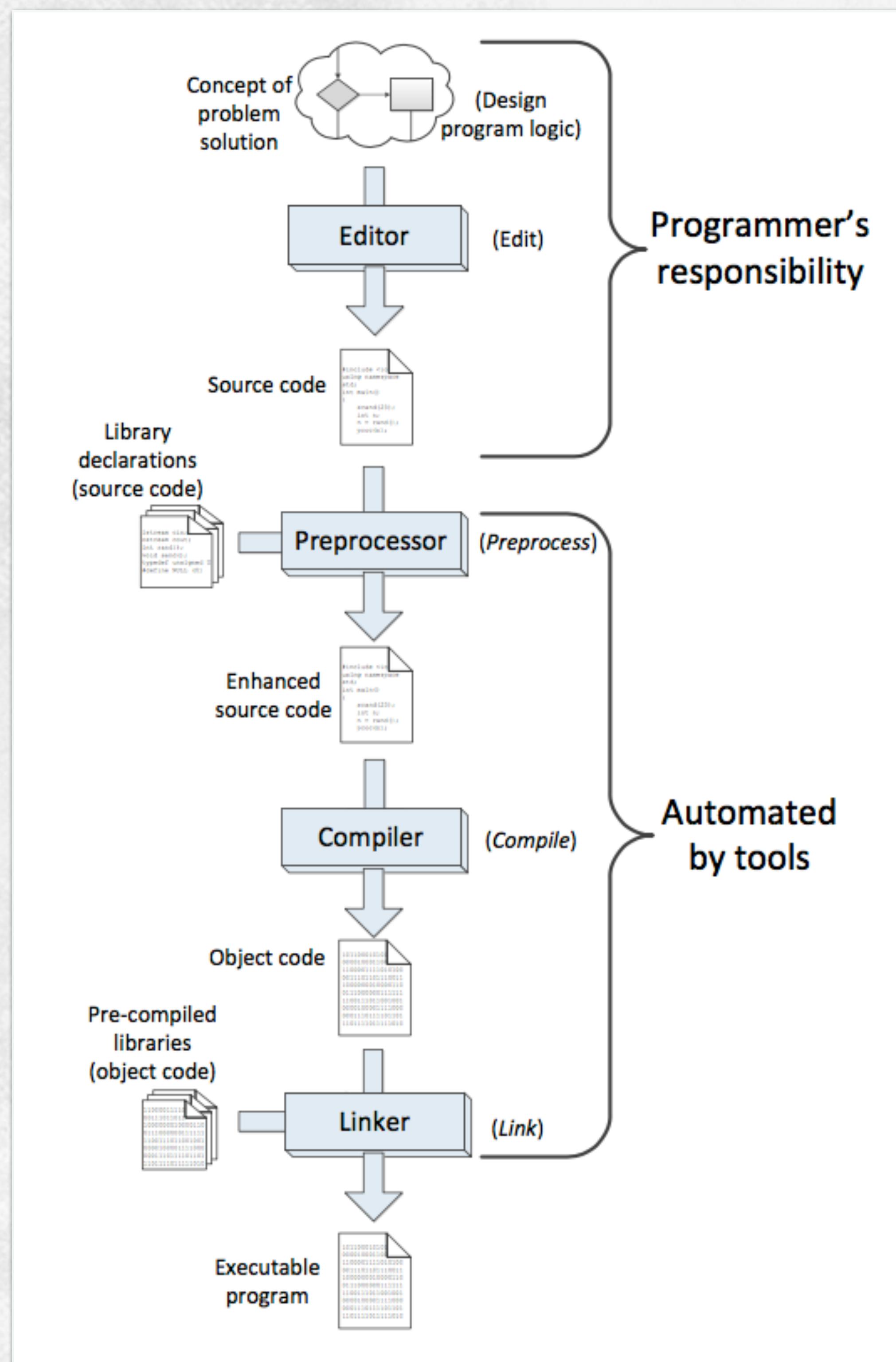
Editor



- An editor allows the user to enter the program source code and save it to files. Most programming editors increase programmer productivity by using colors to highlight language features.
- The syntax of a language refers to the way pieces of the language are arranged to make well-formed sentences. To illustrate, the sentence
- Programmers must follow strict syntax rules to create well-formed computer programs. Only well-formed programs are acceptable and can be compiled and executed. Some syntax-aware editors can use colors or other special annotations to alert programmers of syntax errors before the program is compiled.

Software Build Process & Tools

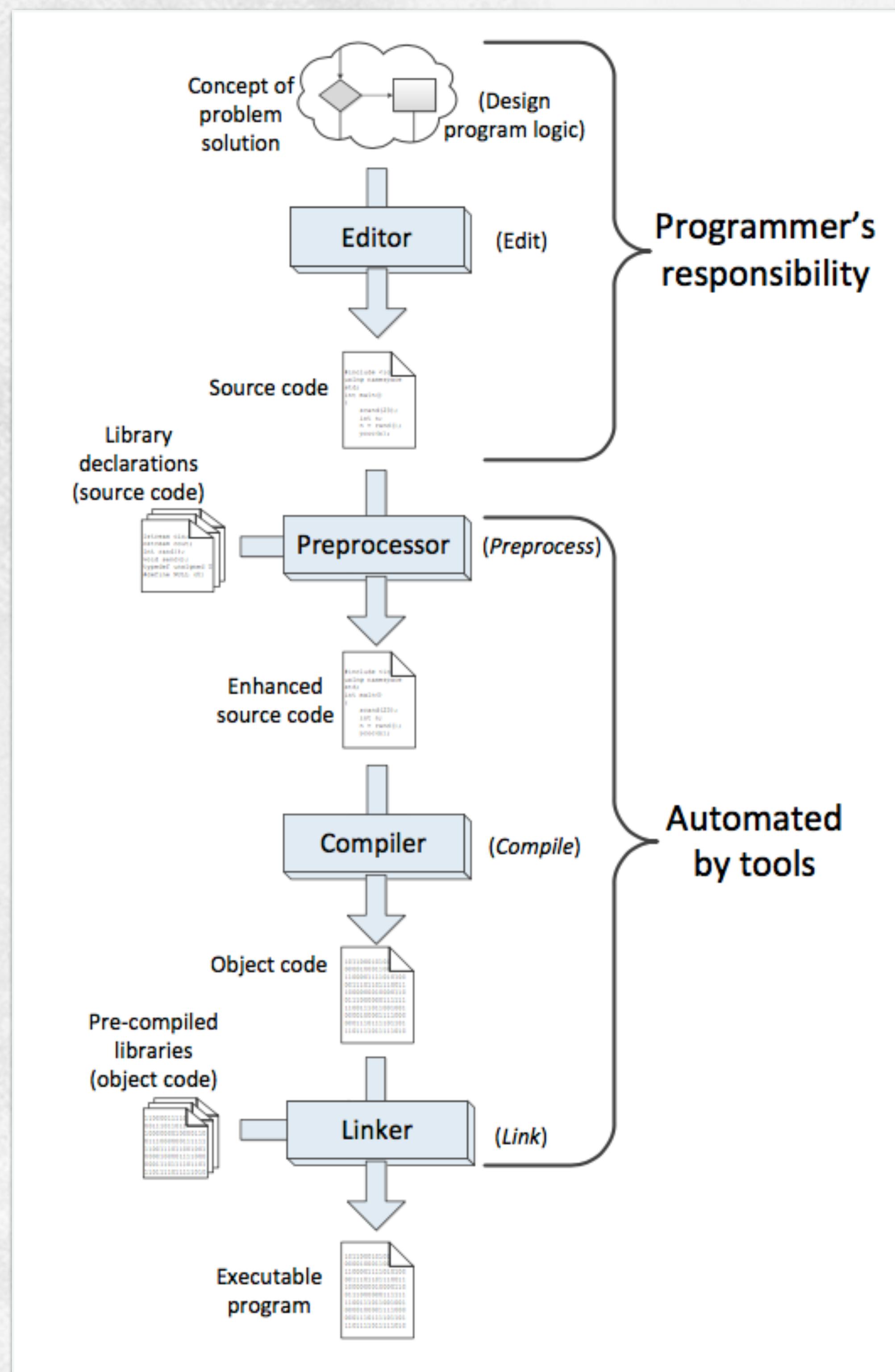
Pre-processor



- Preprocessor adds to or modifies the contents of the source file before the compiler begins processing the code.
- We use the services of the preprocessor mainly to #include information about library routines our programs use.

Software Build Process & Tools

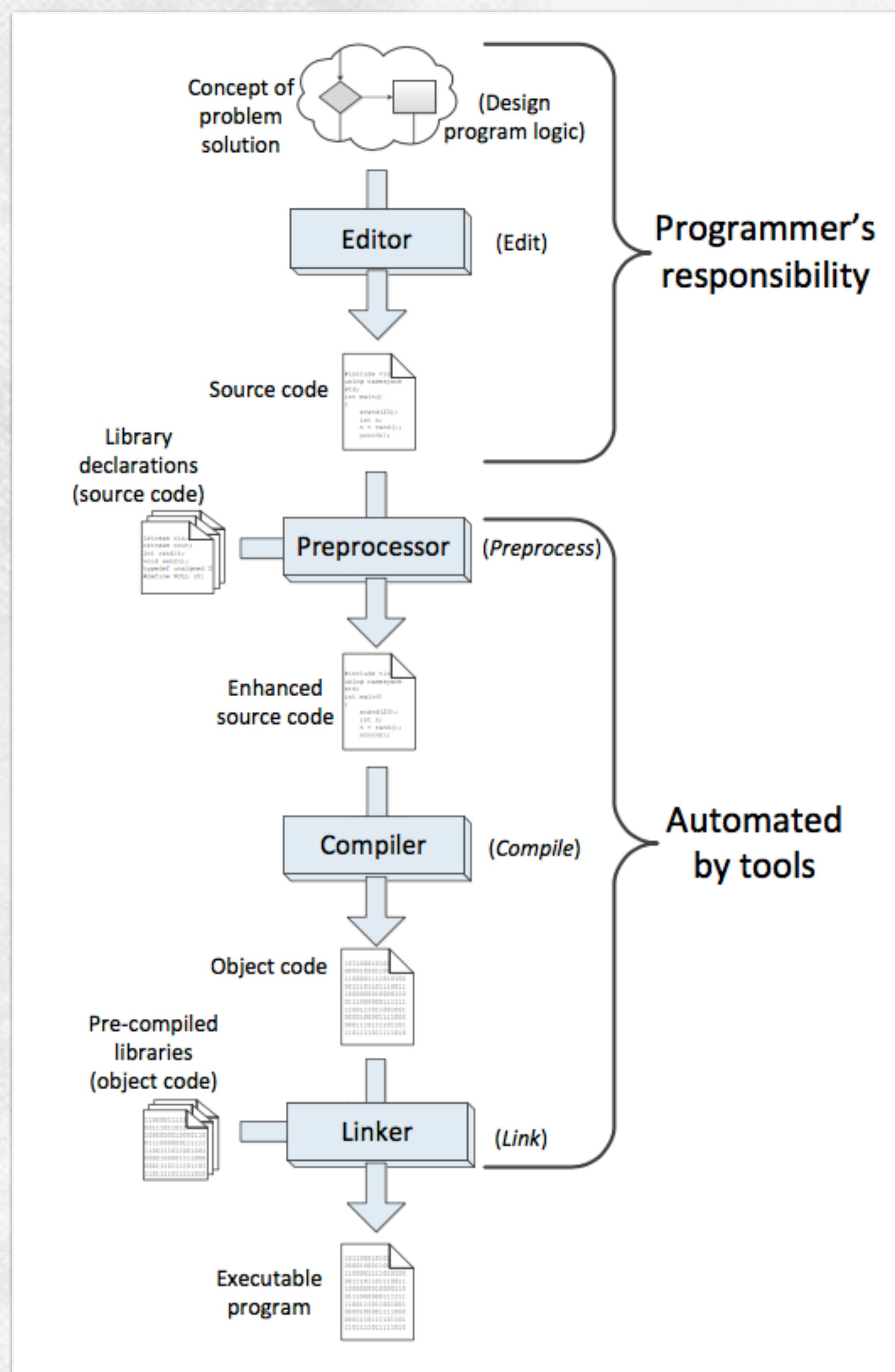
Compiler



- A compiler translates the source code to target code.
- The target code may be the machine language for a particular platform or embedded device.
- The target code could be another source language; for example, the earliest C++ compiler translated C++ into C, another higher-level language.
- The resulting C code was then processed by a C compiler to produce an executable program.
- C++ compilers today translate C++ directly into machine language.

Software Build Process & Tools

Linker



- Linker combines the compiler-generated machine code with precompiled library code or compiled code from other sources to make a complete executable program.
- Most compiled C++ code is incapable of running by itself and needs some additional machine code to make a complete executable program.
- The missing machine code has been precompiled and stored in a repository of code called a library.
- A program called a linker combines the programmer's compiled code and the library code to make a complete program.

Software Build Process & Tools

Debugger

- A debugger allows a programmer to more easily trace a program's execution in order to locate and correct errors in the program's implementation.
- With a debugger, a developer can simultaneously run a program and see which line in the source code is responsible for the program's current actions.
- The programmer can watch the values of variables and other program elements to see if their values change as expected.
- Debuggers are valuable for locating errors (also called bugs) and repairing programs that contain errors. (See Section 4.6 in Textbook for more information about programming errors.)

Software Build Process & Tools

Profiler

- A profiler collects statistics about a program's execution allowing developers to tune appropriate parts of the program to improve its overall performance.
- A profiler indicates how many times a portion of a program is executed during a particular run, and how long that portion takes to execute.
- Profilers also can be used for testing purposes to ensure all the code in a program is actually being used somewhere during testing - this is known as coverage.
- It is common for software to fail after its release because users exercise some part of the program that was not executed anytime during testing.
- The main purpose of profiling is to find the parts of a program that can be improved to make the program run faster.

Famous Free IDEs and Compiler

- IDE: Microsoft Visual C++ 2005 Express Edition [support MS-Windows]

<https://www.microsoft.com/korea/msdn/vstudio/express/visualc/>

- IDE: Microsoft Visual Studio Community [MS-Windows/Mac]

<https://www.visualstudio.com/ko/vs/visual-studio-express/>

- IDE: Xcode [Mac]

<https://developer.apple.com/xcode/>

- IDE: Qt Creator [MS-Windows/Mac/Linux]

<http://doc.qt.io/qtcreator/>

- Compiler: GCC, the GNU C/C++ Compiler [MS-Windows/Mac/Linux]

<https://gcc.gnu.org/>

Software Build Tools & Process

Famous Free Editors

- Editor: Microsoft Visual Studio Code

<https://code.visualstudio.com/>

- Editor: ATOM

<https://atom.io/>

Open Source Resources for Software Programming

KHU Open Sources

- KHU OPEN.FREEDOM Site

<http://opensource.khu.ac.kr>

The screenshot shows the homepage of the KHU OPEN.FREEDOM website. At the top, there's a navigation bar with links for 'Made in KHU', 'ONOS/CORD WG', 'Global Projects', '소식' (News), '행사' (Events), '자료' (Materials), and 'Contact'. Below the navigation is a large banner featuring a building with cherry blossoms and the text 'OPEN.FREEDOM' and '오픈소스를 사랑하는 모든 이를 위한 정보 공유 공간'. To the left of the banner is a section titled 'GLOBAL PROJECTS' with a large logo for 'KHU-Hub' featuring two figures holding bowls and a torch. To the right of the banner is a section titled 'News, Books and Courses' with links to 'opensource.com', 'O'REILLY', and 'Codeacademy'. Below these are sections for 'Programming' with links to various tools and languages like Python, Javascript, Swift, C++, R, GitHub, Git, Eclipse, Node.js, React, CORDOVA, Atom, Geany, and Visual Studio Code.

- KHU-HUB Git Server

<http://khuhub.khu.ac.kr/>

Example Program

Empty

```
int main()
{
}
```

Example Program

Empty

- Line 01:

- This specifies the real beginning of our program. Here we are declaring a function named main.
- All C++ programs must contain this function to be executable. Details about the meaning of int and the parentheses will appear in Week 2. More general information about functions appear in Week 3 and 4.
- The opening curly brace at the end of the line marks the beginning of the body of a function. The body of a function contains the statements the function is to execute.

```
01: int main()
02: {
03: }
```

Example Program

Empty

- Line 02:

- The opening curly brace at the end of the line marks the beginning of the body of a function.
- The body of a function contains the statements the function is to execute.

```
01: int main()
02: {
03: }
```

Example Program

Empty

- Line 03:

- The closing curly brace marks the end of the body of a function.
- Both the open curly brace and close curly brace are required for every function definition.

```
01: int main()
02: {
03: }
```

General Structure of C++ Program

Simple Case

include directives

```
int main() {
```

program statements

```
}
```

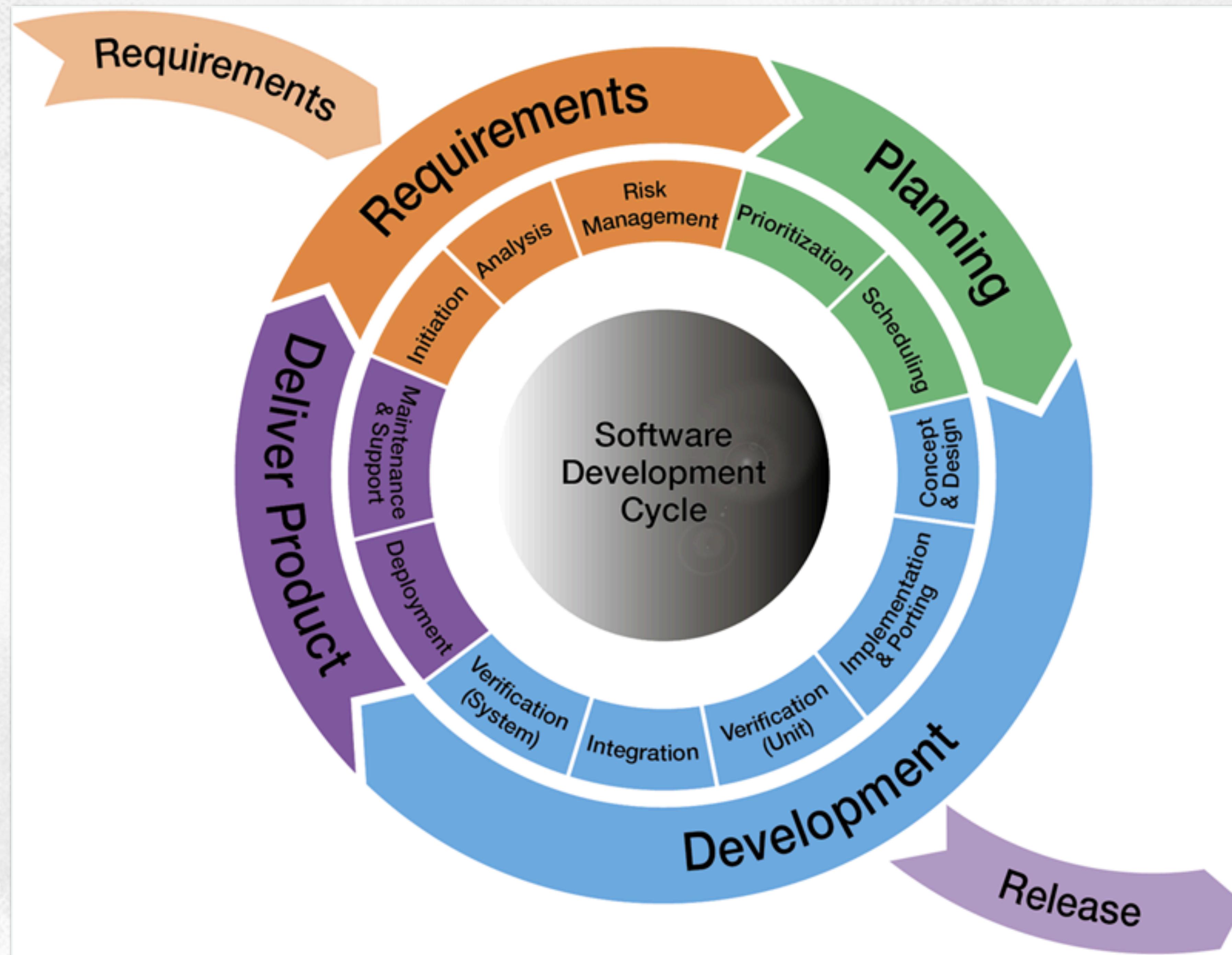
Software Development Stage

Waterfall (for traditional big & large scale projects)



Software Development Stage

Agile (Spiral approach; for fast service oriented projects)





Object Oriented Programming by C++

Sungwon Lee / Professor

Email: drsungwon@khu.ac.kr

Web: <http://mobilelab.khu.ac.kr/>