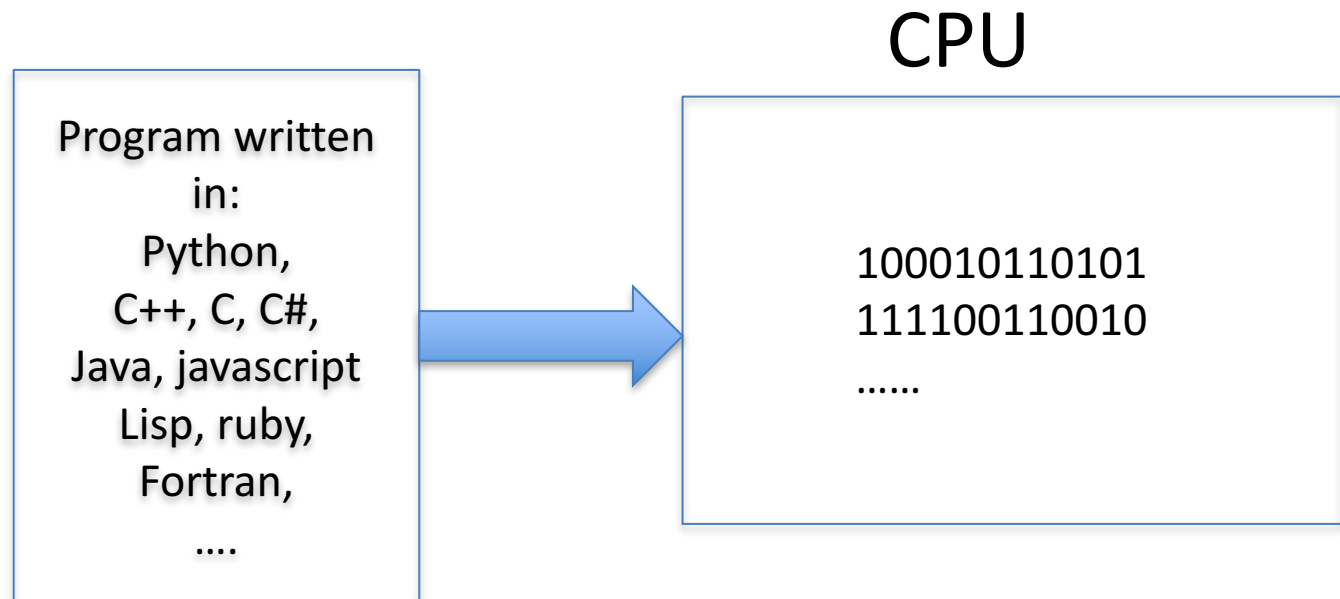


# Python vs. C++

Based on “A Transition Guide: Python to C++”  
by M. Goldwasser and D. Letscher

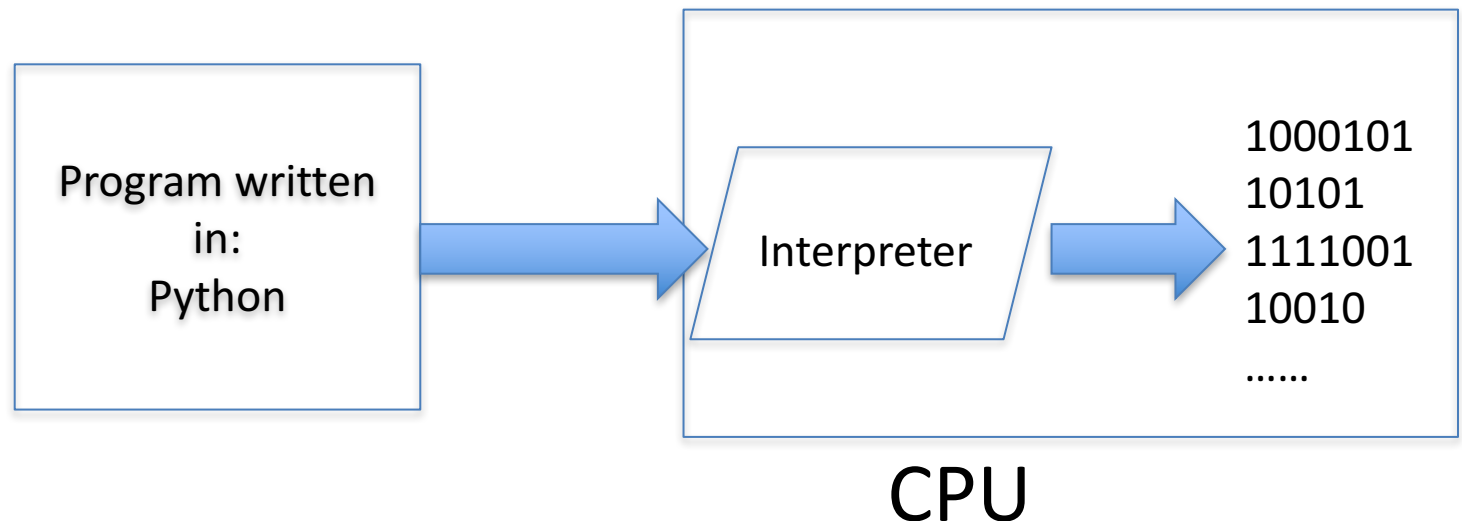
# High Level Languages

- There are more than 1000 high level computer languages developed
- Around one hundred actively used for program development



# Interpreter vs. Compiler

- Python is an interpreted language
  - Use a Python interpreter
- An **interpreter** translates the high level language to low-level language operations **on-the-fly**, **each time** the program is run

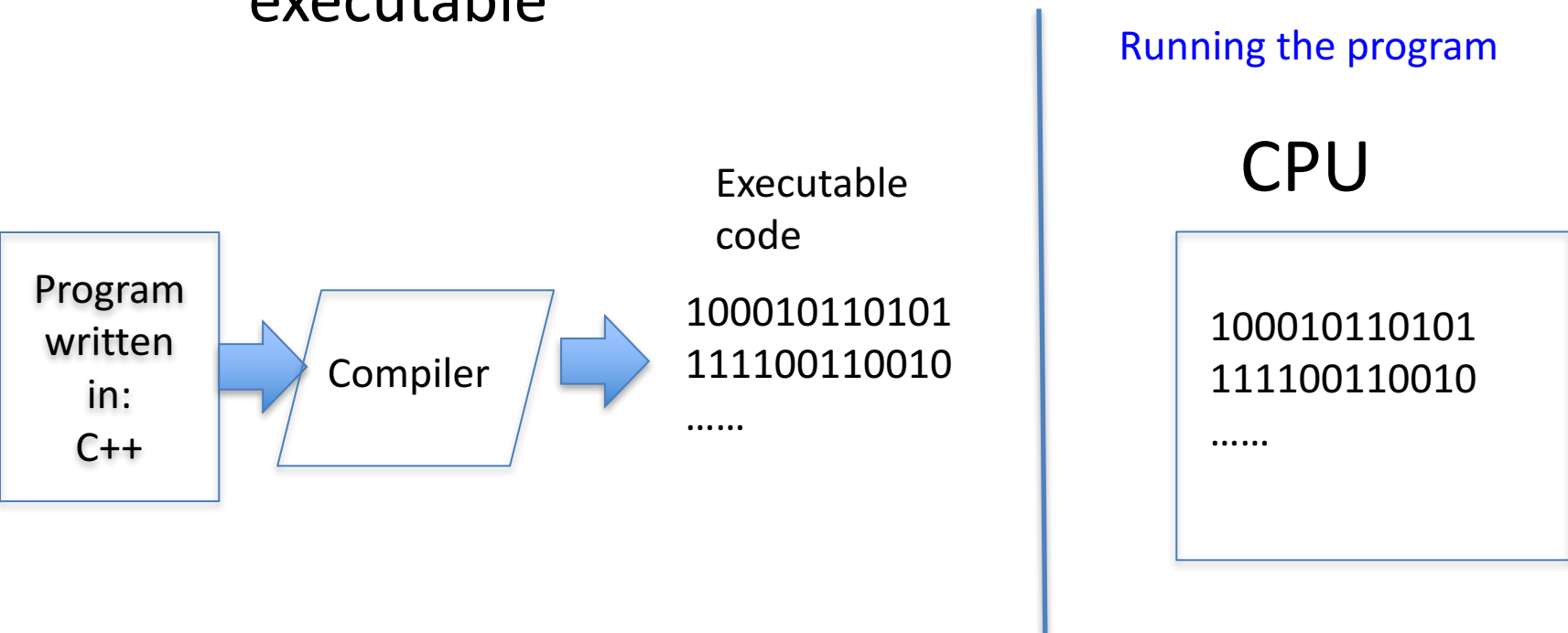


# Interpreter vs. Compiler

- C++ is a compiled language
  - Need a C++ compiler
- A **compiler** analyzes the source code based on the syntax of the language. If there are syntax errors, they are reported and the compilation fails. Otherwise, the compiler translates the high-level code into machine code for the computing system, generating another file known as an **executable**.

# Interpreter vs. Compiler

- Two step process for C++
  - Stage one: Compile – if successful → executable
  - Stage two: Run the program by running the executable



# Interpreter vs. Compiler

- What are the implications of this difference ?

# Interpreter vs. Compiler

- What are the implications of this difference ?
  - Computation speed
  - Software distribution
  - Platform independence
  - Debugging cycle

# Dynamic vs. Static Typing

- Dynamically typed language (Python)
  - For example, within a given scope, an identifier may be assigned values of different types:

```
age = 38  
...  
age = "Nice work!"
```
  - Function parameters serves as place holders, no types specified

```
def ComputeSum(v1, v2):
```
  - The validity of functions such as `age.lower()` is determined at interpretation time (run time)



# Dynamic vs. Static Typing

- Statically Typed Language (C++)
  - An explicit type declaration is required for each identifier before it can be used:

```
int age;  
age = 21;
```

- Same applies to the function parameters  
`float ComputeSum(float v1, float v2)`
- Type checks are done during compile time

# Dynamic vs. Static Typing

- What are the implications of this difference?

# Dynamic vs. Static Typing

- What are the implications of this difference?
  - Early detection of error (static typing)
  - Reduced syntactical burden and simpler support for polymorphism (dynamic typing)

# Choice of Python vs. C++

- C++ is among the most widely used coding languages in industry
- It can create fast executables and robust libraries
- It allows the programmers to control low-level aspects of :
  - How data is stored
  - How information is passed, and
  - How memory is managed

# Choice of Python vs. C++

- Python is a good programming language for fast prototyping
  - Figure out the needed functionalities of a system

# Main weakness of C++

- Complexity
  - Large number of libraries, functions, and features have been built over the year.
  - There are often many ways of doing one thing
    - Some are efficient ways and others may be less efficient, or even flawed