CptS 355- Programming Language Design

Implementing Programming Languages

Java Virtual Machine

Instructor: Sakire Arslan Ay

WASHINGTON STATE UNIVERSITY

World Class. Face to Face.

Implementing Programming Languages

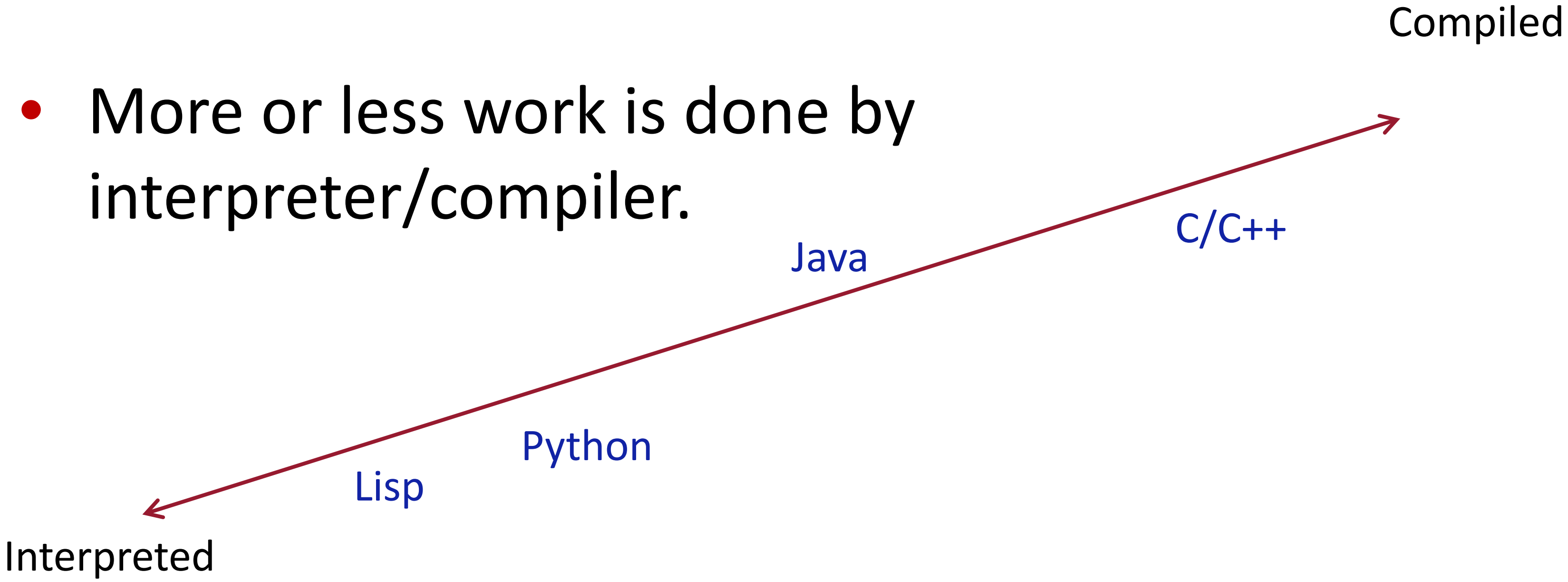
- How to implement programming models?
  - Compiled languages:
    - Examples: C, C++, Fortran, Pascal, Haskell
  - Interpreted languages:
    - Examples: LISP, Scheme, Python, Matlab, Perl

Implementing Programming Languages

- Advantages of Interpreted Languages
  - Execute line by line in original source code
  - Easier to program and debug
    - Un-typed variables
    - On the fly variable creation
  - Easier to run on different architecture:
    - Runs as a simulated environment that exists inside the interpreter process.
  - Less work for compiler – all work done at run time
- Disadvantages of Interpreted Languages
  - Much slower to execute
  - Might be ineffective for large scale applications

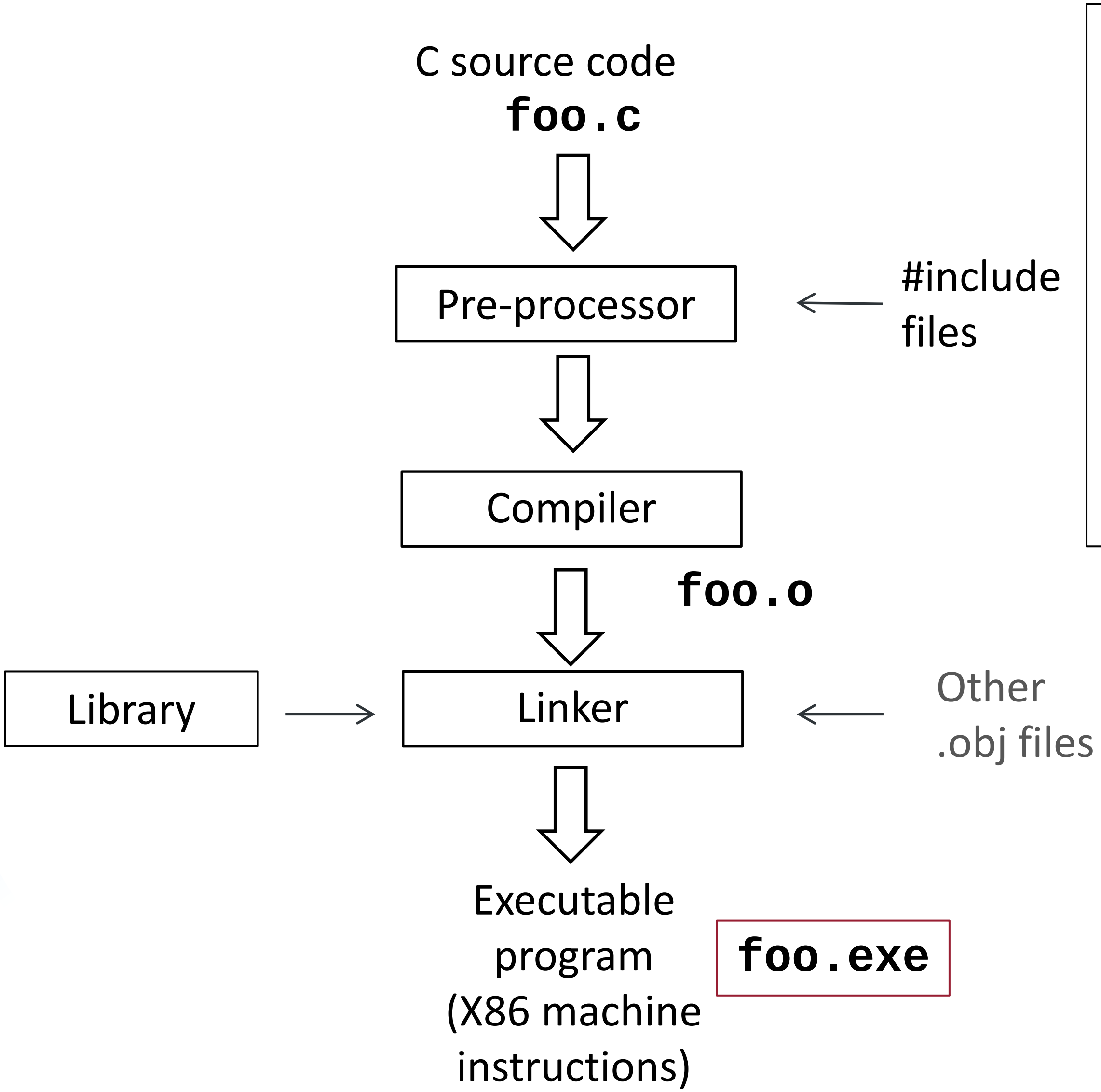
Interpreted vs. Compiled

- More or less work is done by interpreter/compiler.



- Java programs are usually run by a virtual machine
  - VMs interpret an intermediate, “partly compiled” language called bytecode.

The C Programming System



C source code
foo.c

Pre-processor

#include files

Compiler

foo.o

Library

Linker

Other .obj files

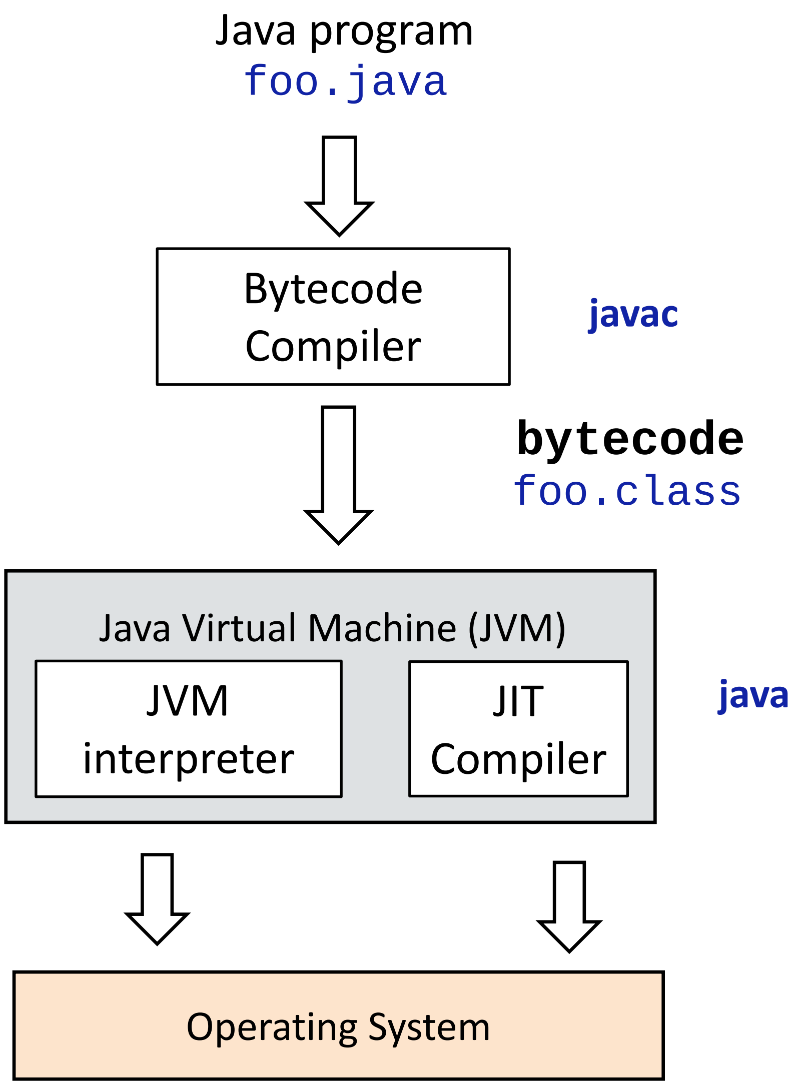
Executable program (X86 machine instructions)

foo.exe

- C is an example of a compiled language
- `gcc` is a script which hides steps
  - `gcc -c foo.c`
    - creates `foo.o`
  - `gcc -o foo foo.o`
    - links/creates `foo.exe`

Physical Machine

Virtual Machine Model



- Java Virtual Machine
  - Makes the Java language machine-independent
  - Provides strong protections
  - Stack based execution model
  - There are many JVMs
    - Some interpret
    - Some compile into assembly
    - Usually implemented in C

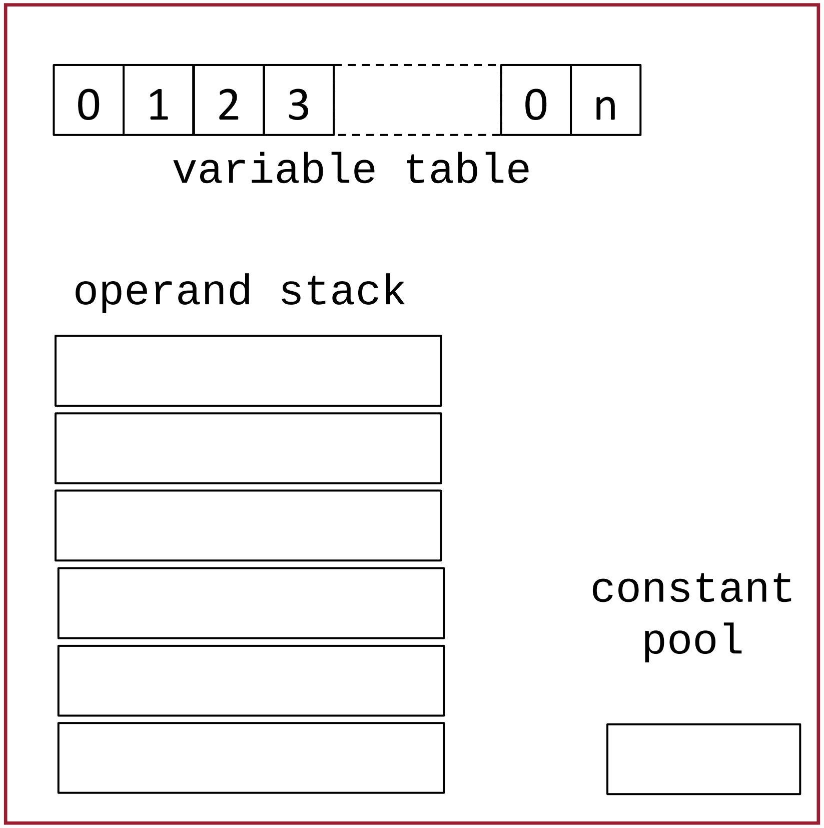
Java Bytecodes

```
iload 1      //push 1st argument from table onto stack
iload 2      //push 2nd argument from table onto stack
iadd        //pop top 2 elements from stack
iload 2
```

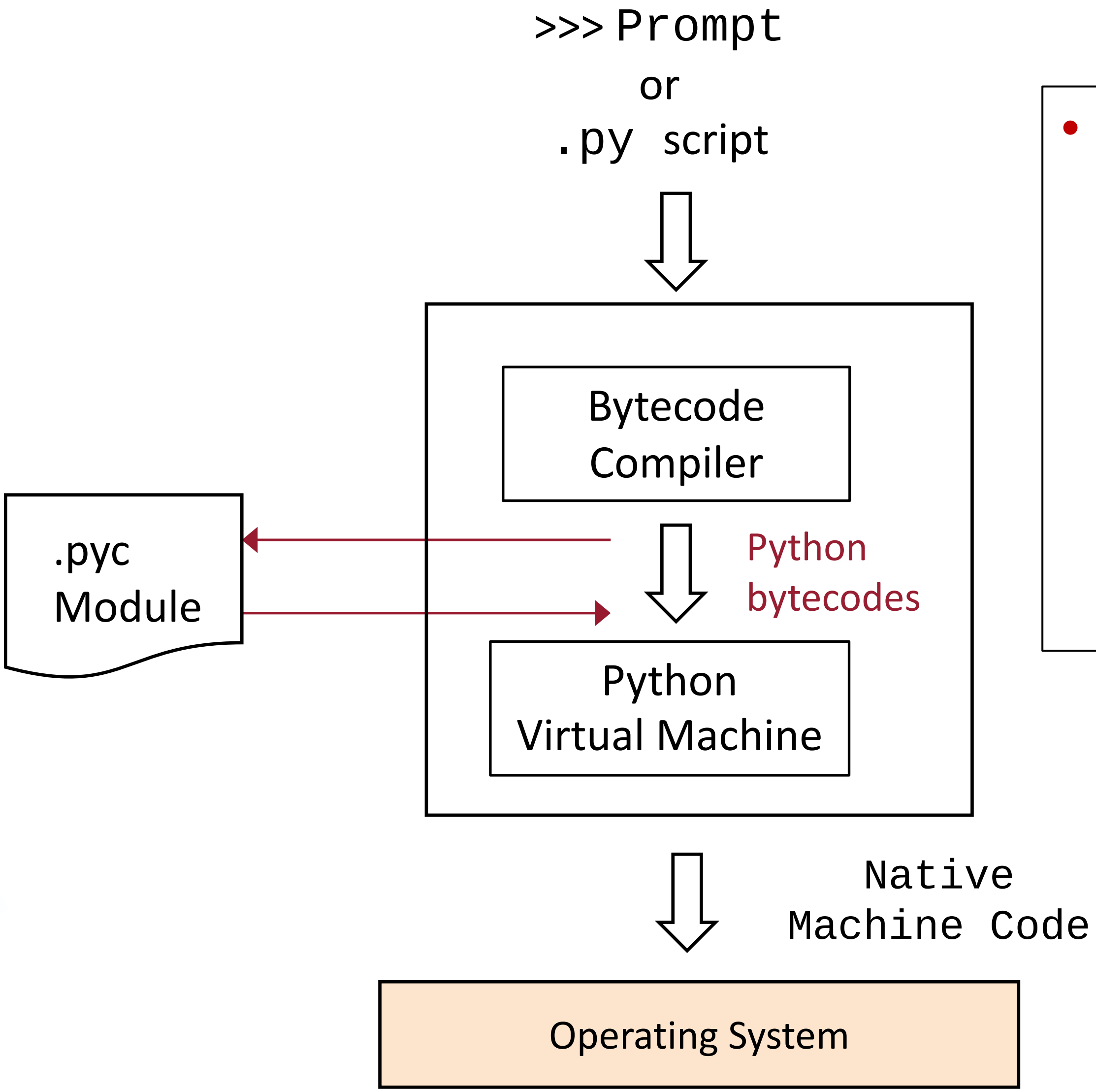
‘i’ stands for integer
‘a’ for reference
‘b’ for byte
‘c’ for char
‘d’ for double

No knowledge of integers memory locations (each instruction is 1 byte – bytecode)

```
mov 0x8001, %eax
mov 0x8002, %edx
add %edx, %eax
mov %eax, 0x8003
```

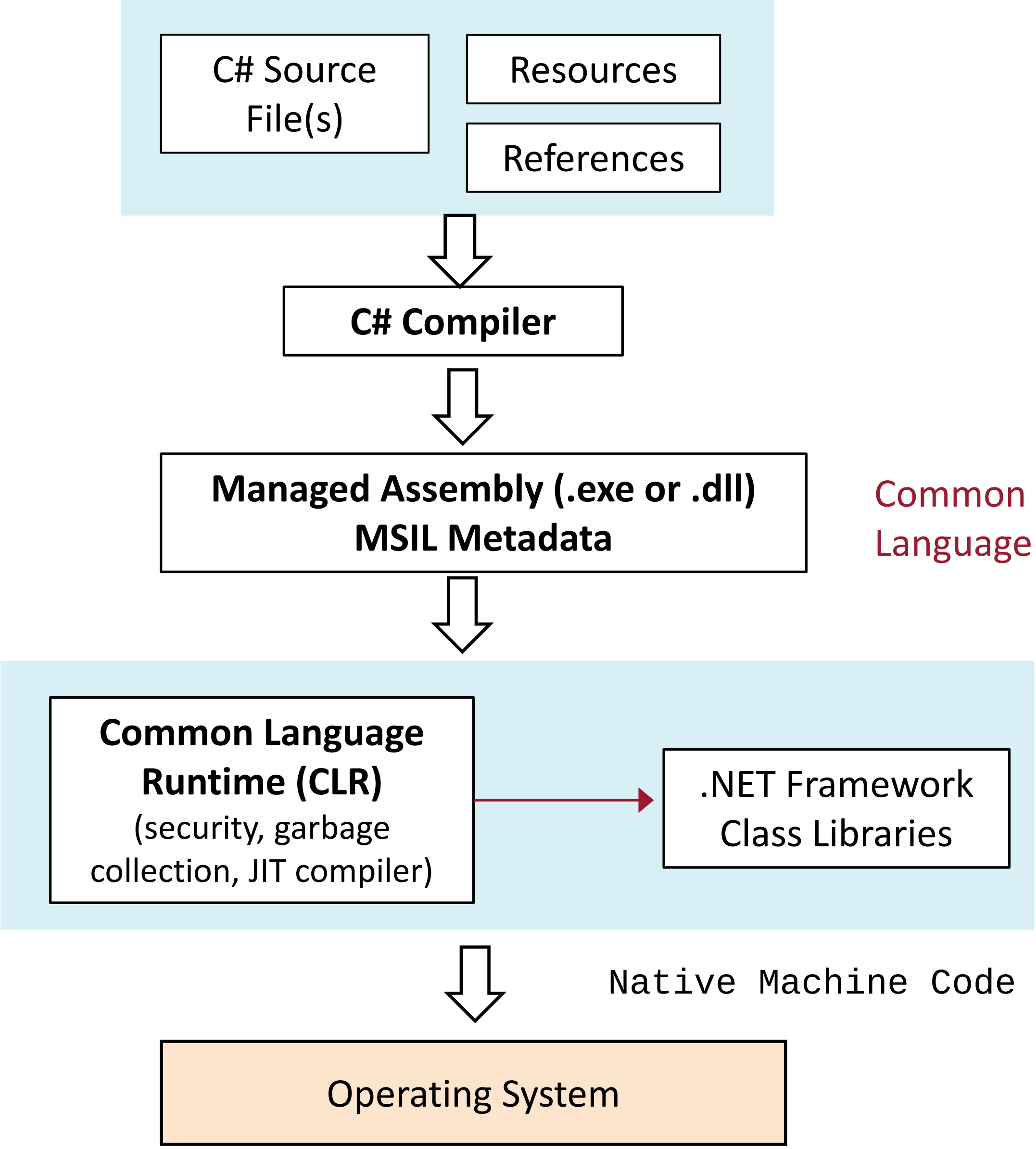


Python Interpreter



- The Python interpreter consists of two parts
  - A Python bytecode compiler
  - A virtual machine which executes Python

C# and .NET Framework Platform



C# Source File(s)

Resources

References

C# Compiler

Managed Assembly (.exe or .dll)
MSIL Metadata

Common Intermediate Language (CIL)

Common Language Runtime (CLR)
(security, garbage collection, JIT compiler)

.NET Framework Class Libraries

Native Machine Code

Operating System