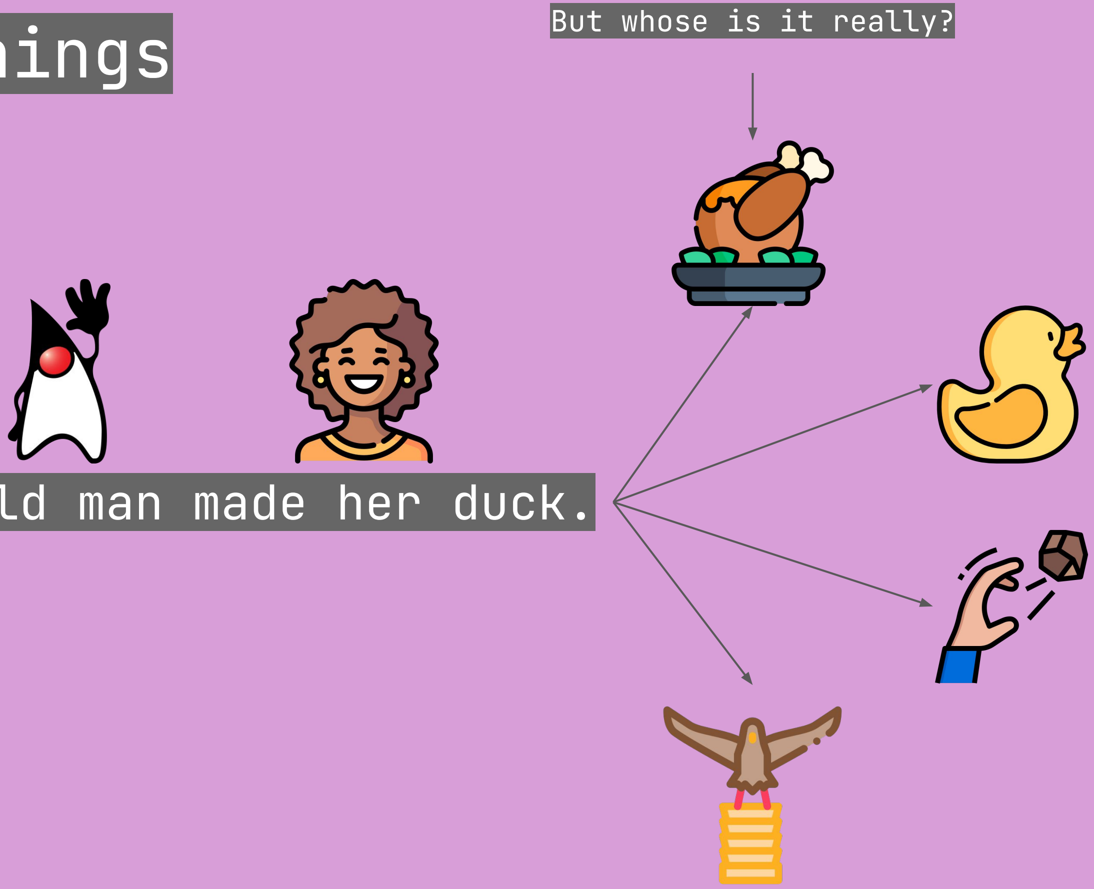
CMPSC 201: PROGRAMMING LANGUAGES

Natural language

At your tables, come up with what you think this sentence might mean:

The old man made her duck.

Possible meanings



The old man made her duck.

But whose is it really?

Natural language

The man who hunts ducks out on weekends.

The sour drink from the ocean.

The old man the boat.

The raft floated down the river sank.

Iron people eat accumulates.

?

Natural language vs. Formal language

Natural language	**Formal language**
Purpose: Communication	Purpose: Computation
Not designed; evolved	Designed
Ambiguous	Unambiguous
Redundant	Concise
Mostly *not* literal	Extremely literal

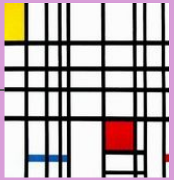
Programming languages are formal

Needs to be:

- Verifiable
- Transferrable
- Completely literal/translatable
- Correct
- Deterministic

Reasons for designing a language

- Suits personal or professional use (preference)
- Solves a specific kind of problem very well
- Existing language deficiencies inspire new replacements
- Inherently rewarding task
- Exploring new computational terrain
- You have a v. funny joke



Nope.

Lox is a “hobby language”

- Created by Robert Nystrom solely as teaching tool
- Adopts features of many languages you might already know:
  - “Duck” typing
  - Brace scoping
  - Object-oriented features:
    - Classes
    - Inheritance
- Two official (more-or-less) implementations:
  - jlox (Lox Java interpreter)
  - clox (Lox C runtime)

jlox

We’re spending the semester writing Nystrom’s implementation of jlox using a single repository, with branches dedicated to each chunk of content.

Sample Lox “Hello, World!”

```
fun sayHello() {
  print “Hello, World!”;
}

sayHello();
```

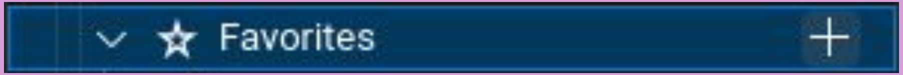
Write this in the test.lox file located in:

…lox/interpreter/src/test/resources

Set up a Favorite for java:exec

Under the “Interpreter” entry in the “Lox Language”:

- Click the + next to “Favorites”

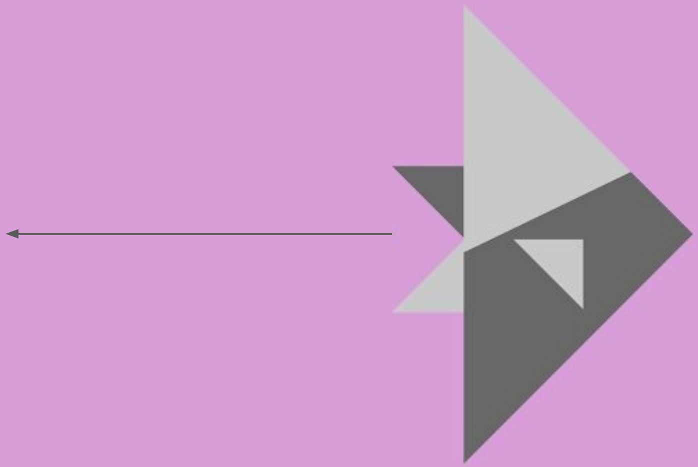


- In the resulting text box, type:

```
exec:java -q
```

Goin’ Fishin’

```
lox
loxlox
loxloxlox
loxloxloxlox
loxloxloxloxlox
loxloxloxlox
loxloxlox
loxlox
lox
```



Write this in the test.lox file located in:

…lox/interpreter/src/test/resources

Interpreted vs. Compiled languages

Interpreted		Compiled
Translated to machine code line by line during execution	PROCESS	Translated to machine code all at once before execution
Slower execution speed: line-by-line processing	SPEED	Faster execution speed: machine code is available at once
Platform independent as long as there’s an interpreter version	DEPENDENCY	Machine dependent; has to be compiled for target system(s)
Programmers debug at runtime, enabling faster decision-making	ERRORS	Compilers raise errors before program can even build
?	USE CASES	?
?	EXAMPLES	?

Interpreted vs. Compiled languages

Interpreted

Both translates and runs code

Compiled

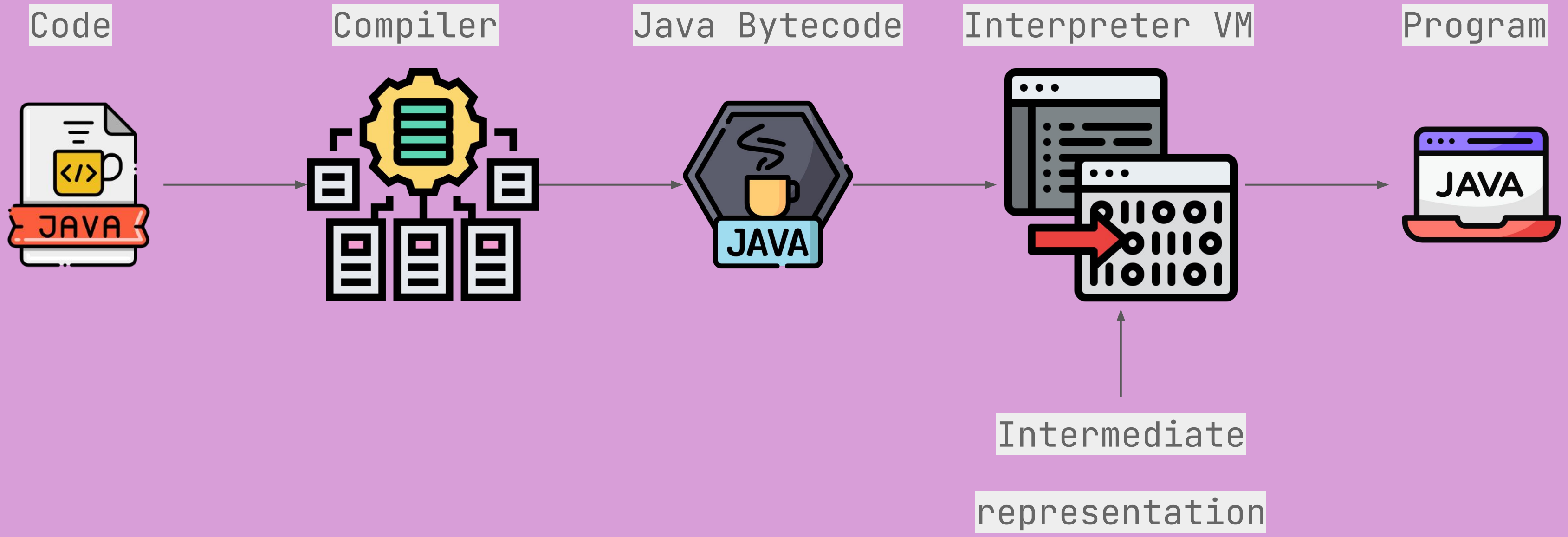
Only builds code; doesn’t run

Hybrid

Builds an *intermediate version*, the runs it in a *virtual machine*

Is portable *and* performant

Hybrid approach in detail



Code

Compiler

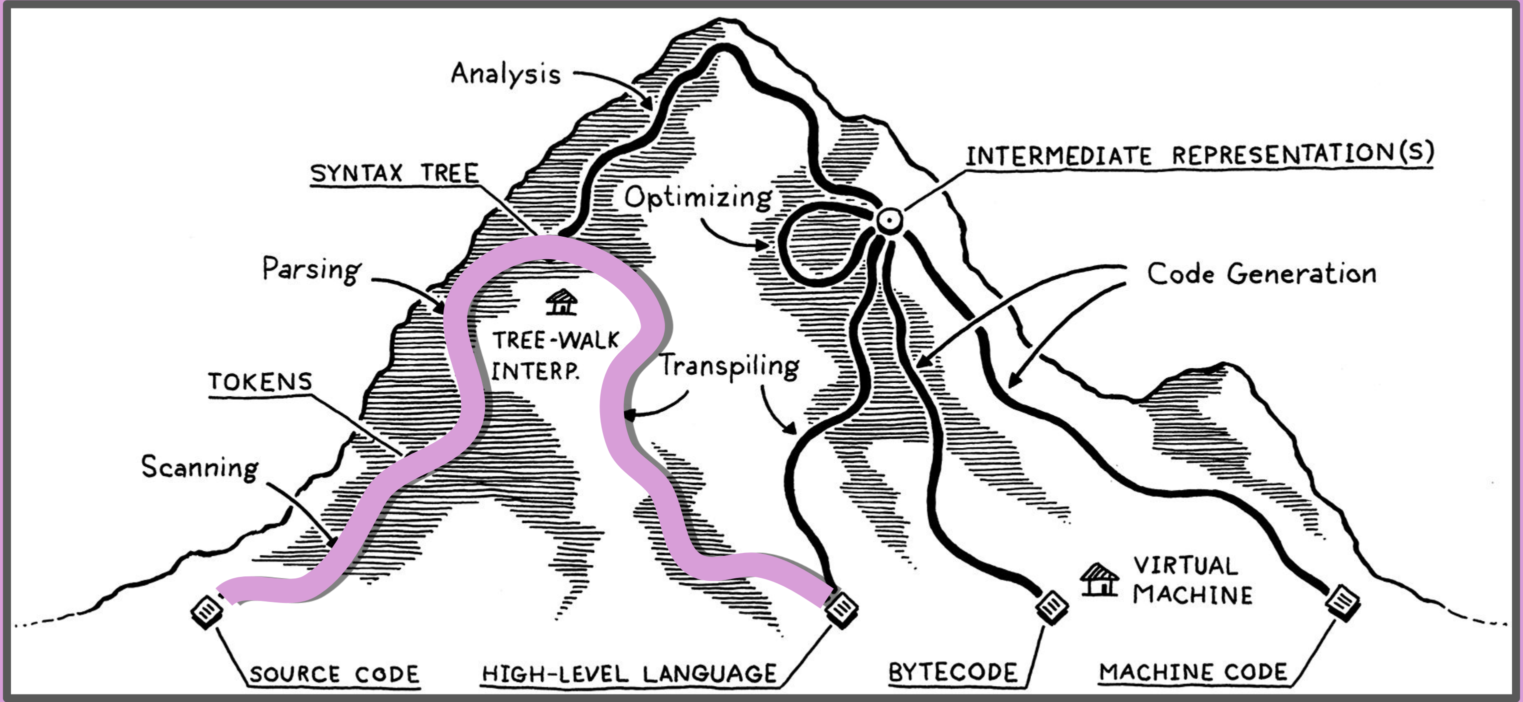
Java Bytecode

Interpreter VM

Program

Intermediate

representation


Analysis
SYNTAX TREE
Parsing
TOKENS
Scanning
TREE-WALK INTERP.
Optimizing
Transpiling
INTERMEDIATE REPRESENTATION(S)
Code Generation
VIRTUAL MACHINE
SOURCE CODE
HIGH-LEVEL LANGUAGE
BYTECODE
MACHINE CODE