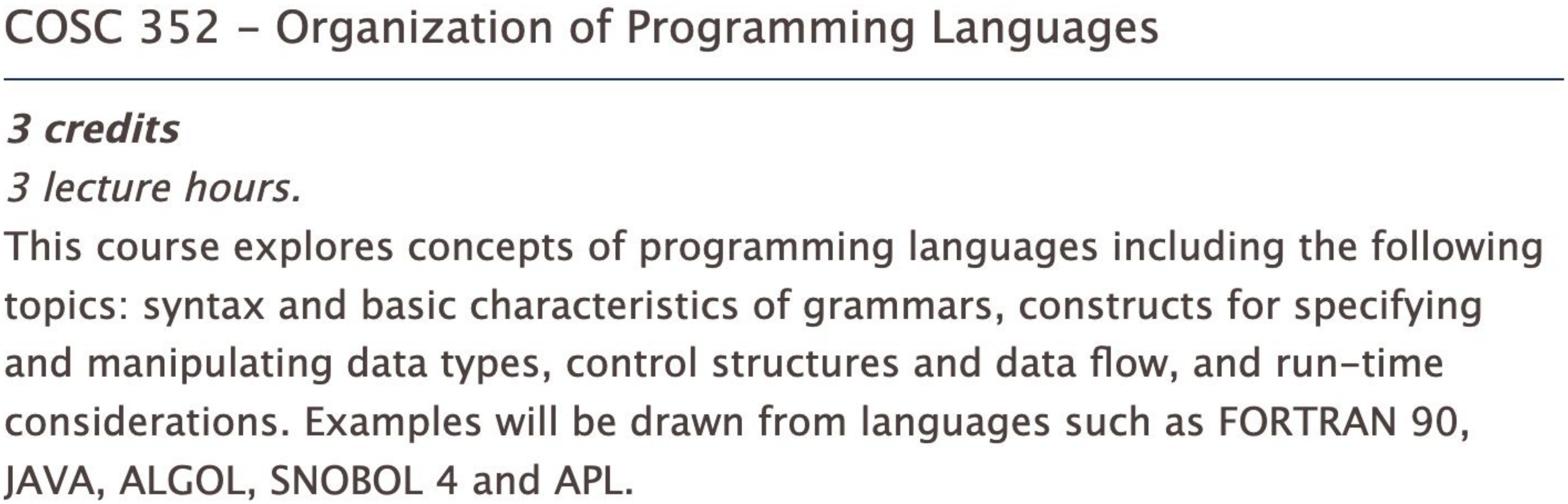
Organization of Programming Languages (COSC 352)

Dr Paul Wang

About Me (Jon White)

- Graduated Computer Science 2001
- Heavily involved in SACS
- Started work at Lockheed Martin
- Been in DefTech for 20+ years
- Currently a Software Architect

About This Class



COSC 352 – Organization of Programming Languages

3 credits

3 lecture hours.

This course explores concepts of programming languages including the following topics: syntax and basic characteristics of grammars, constructs for specifying and manipulating data types, control structures and data flow, and run-time considerations. Examples will be drawn from languages such as FORTRAN 90, JAVA, ALGOL, SNOBOL 4 and APL.

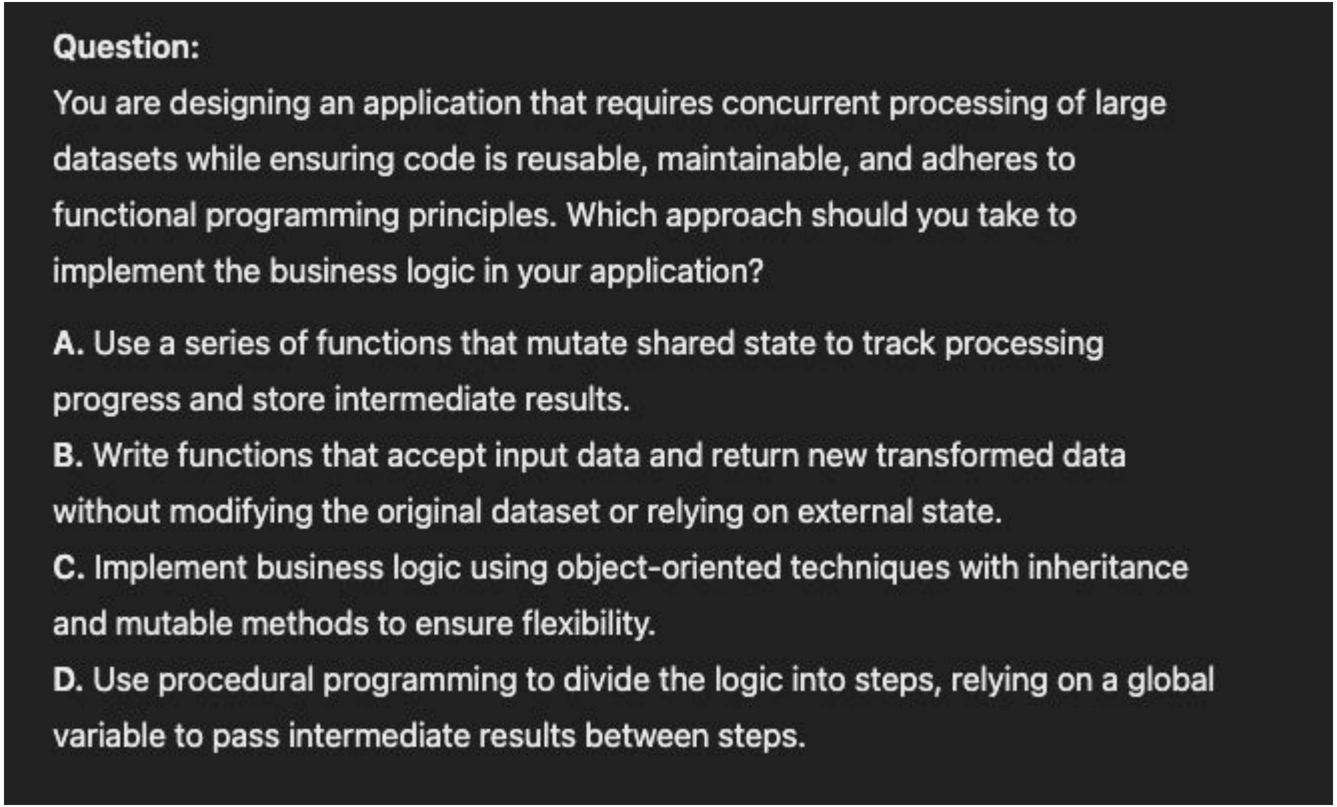
About Me

- 5 Projects - 50%
- 4 In-Person Tests Certification Style - 20%
- Presentations for Mid-Term - 15%
- Comprehensive Final Exam - 15%

Projects (50%)

- Straightforward
- Expose you to a new language
- Free to use GenAI
- Can only use native language features
- Will have to follow a strict format

Certification Style Tests (20%)



Question:

You are designing an application that requires concurrent processing of large datasets while ensuring code is reusable, maintainable, and adheres to functional programming principles. Which approach should you take to implement the business logic in your application?

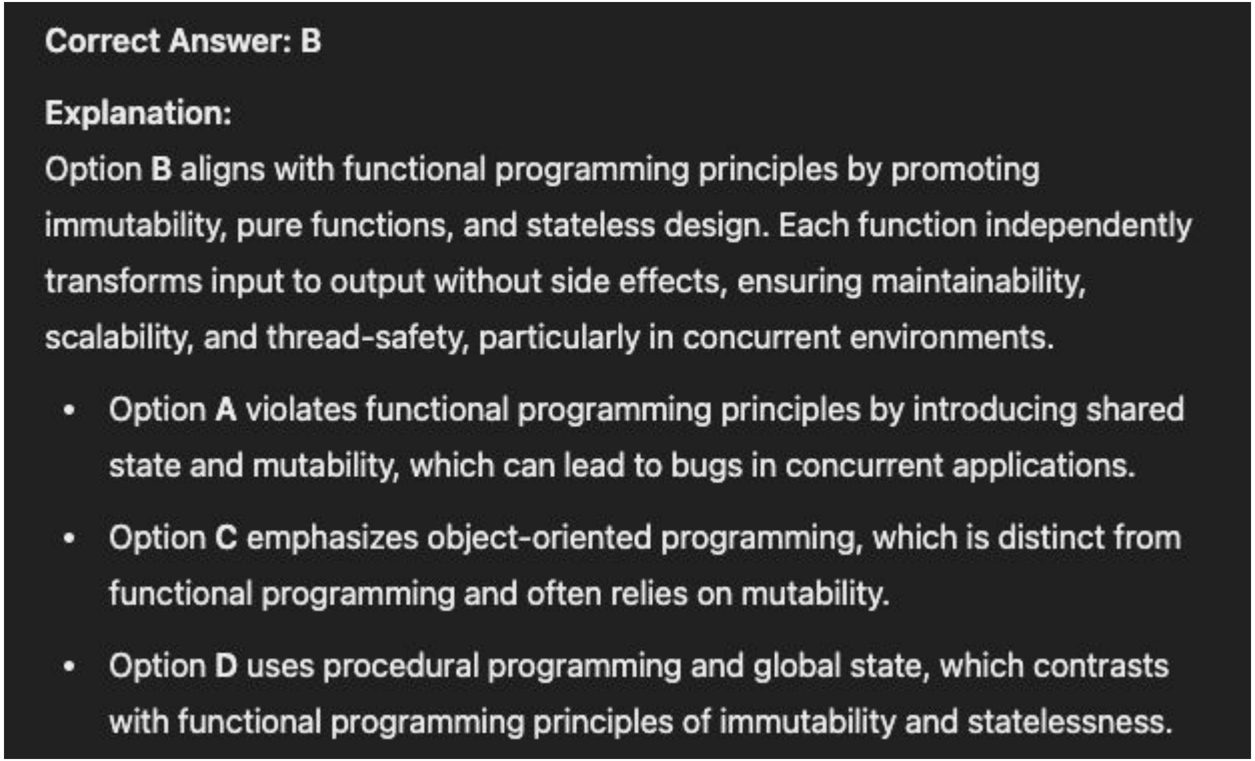
A. Use a series of functions that mutate shared state to track processing progress and store intermediate results.

B. Write functions that accept input data and return new transformed data without modifying the original dataset or relying on external state.

C. Implement business logic using object-oriented techniques with inheritance and mutable methods to ensure flexibility.

D. Use procedural programming to divide the logic into steps, relying on a global variable to pass intermediate results between steps.

Certification Style Tests



Correct Answer: B

Explanation:
Option **B** aligns with functional programming principles by promoting immutability, pure functions, and stateless design. Each function independently transforms input to output without side effects, ensuring maintainability, scalability, and thread-safety, particularly in concurrent environments.

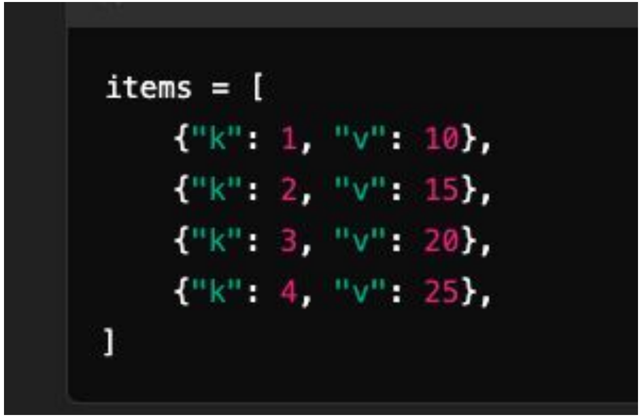
- Option **A** violates functional programming principles by introducing shared state and mutability, which can lead to bugs in concurrent applications.
- Option **C** emphasizes object-oriented programming, which is distinct from functional programming and often relies on mutability.
- Option **D** uses procedural programming and global state, which contrasts with functional programming principles of immutability and statelessness.

Mid-Term Presentation (15%)

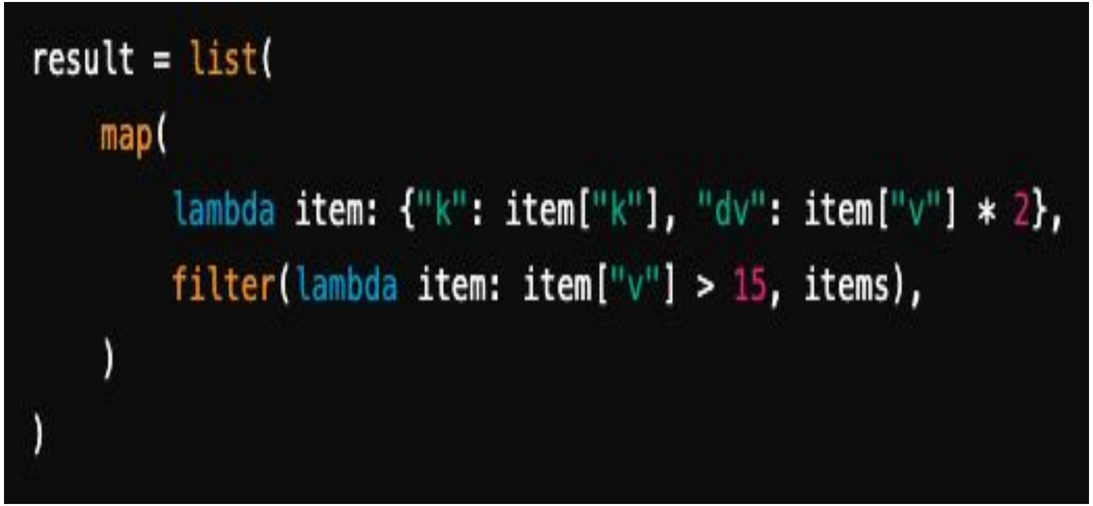
- In front of class
- 10-15 Minutes
- Teach the class about language features
- Demonstration of Program or code walkthrough
- Class will participate in grading

Final (15%)

- Focus on applied concepts of polyglot programming
- What programming paradigm is featured here?



```
items = [
    {"k": 1, "v": 10},
    {"k": 2, "v": 15},
    {"k": 3, "v": 20},
    {"k": 4, "v": 25},
]
```



```
result = list(
    map(
        lambda item: {"k": item["k"], "dv": item["v"] * 2},
        filter(lambda item: item["v"] > 15, items),
    )
)
```

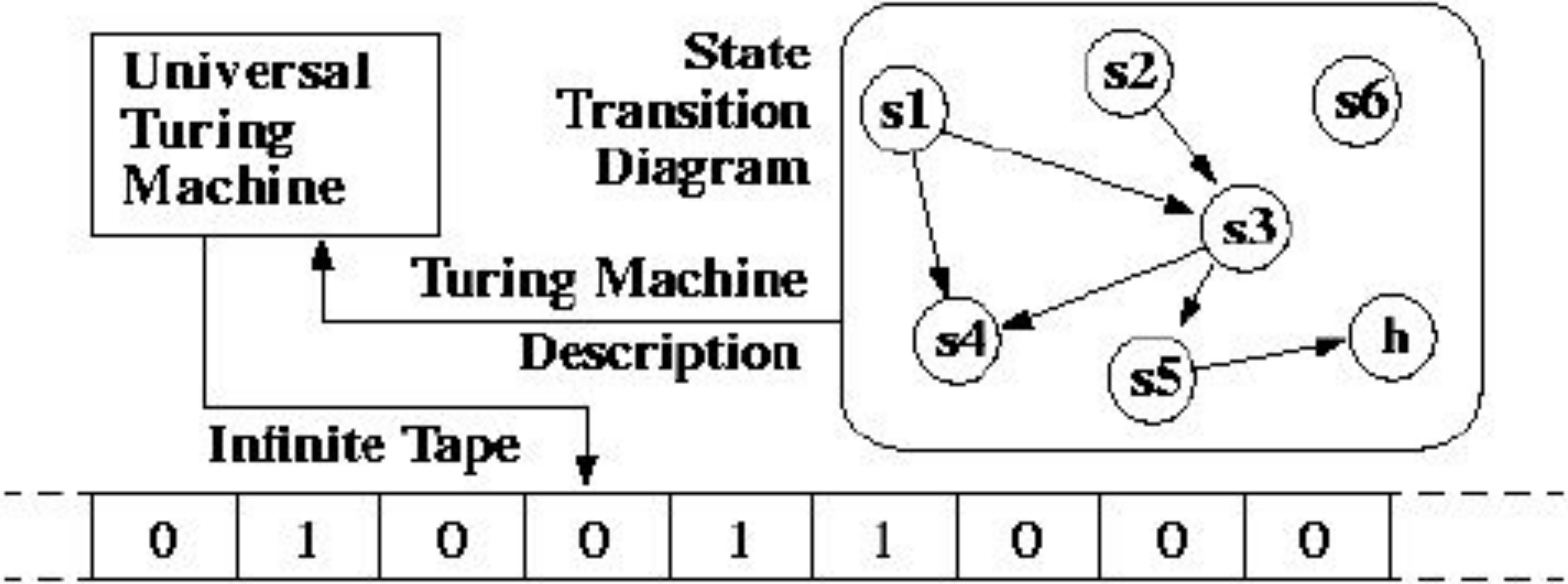
Course Outline

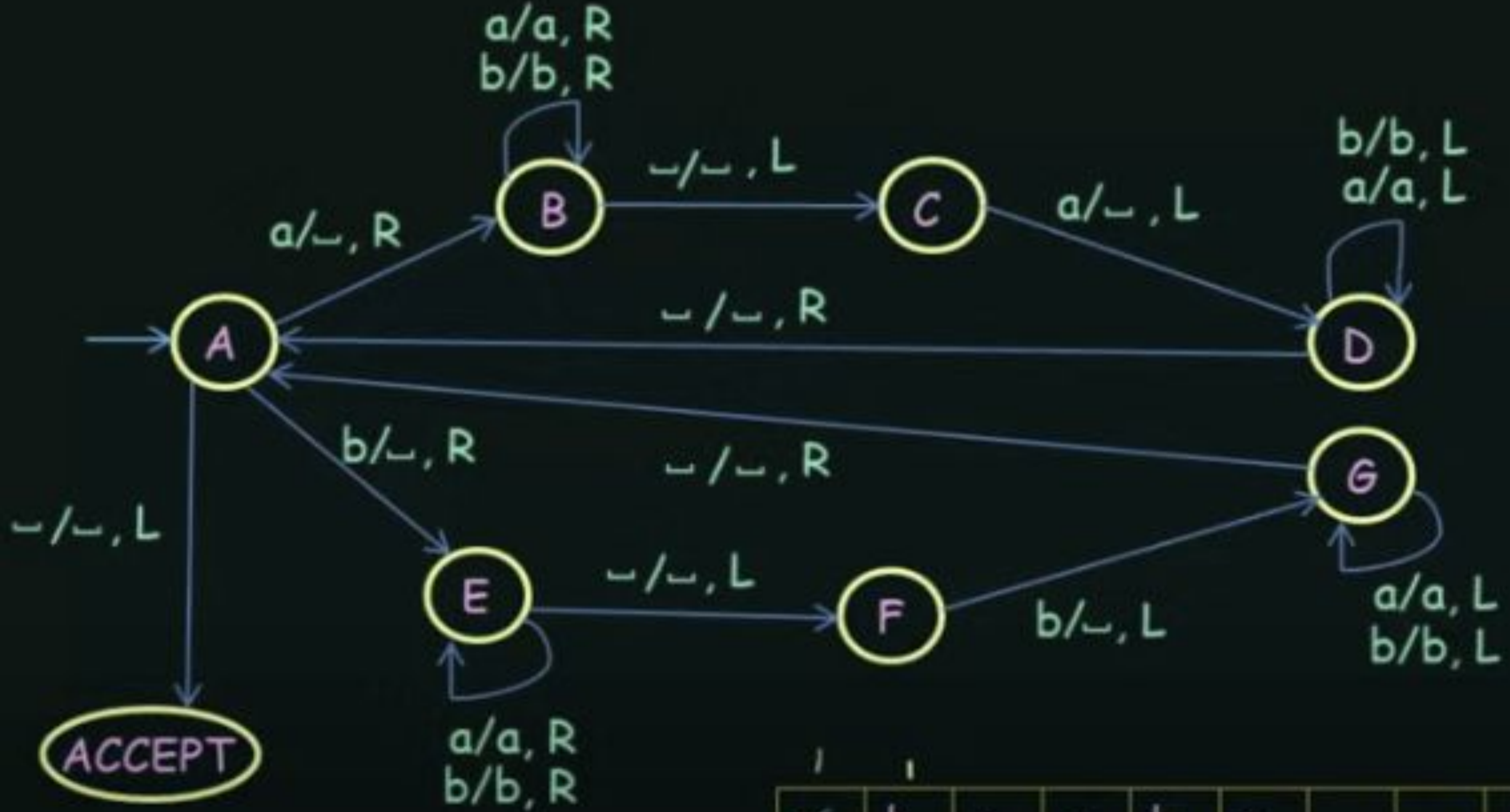
What is a polyglot?

What is a polyglot in reference to software developer?

Why is being proficient in multiple languages important?

How will ChatGPT effect software development in general?


Universal Turing Machine
State Transition Diagram
Turing Machine Description
Infinite Tape
s1
s2
s6
s3
s4
s5
h
0 1 0 0 1 1 0 0 0


a/a, R
b/b, R
B
␣/␣, L
C
a/␣, L
b/b, L
a/a, L
D
a/␣, R
A
␣/␣, R
b/␣, R
␣/␣, R
G
␣/␣, L
E
␣/␣, L
F
b/␣, L
a/a, L
b/b, L
a/a, R
b/b, R
ACCEPT

~~a~~	b	a	a	b	a	␣	␣	....

What Does Turing Complete Mean?

Key characteristics that define a Turing complete system

Sequence

Execute a series of computational steps in the order they are provided.

Conditionals

Execute different computational paths conditionally based on certain criteria.

Iteration

Repeat computational sub-processes over and over.

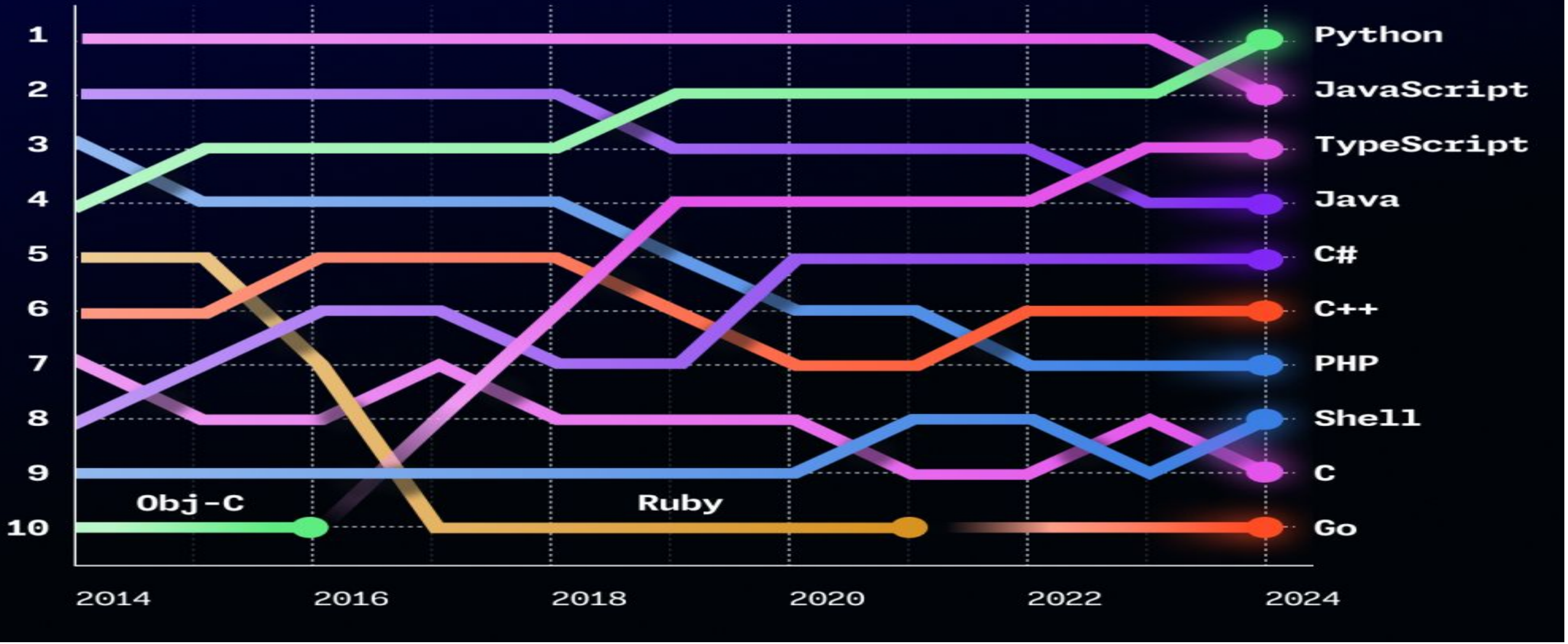
Store Data

Store intermediate results for later use in memory.

@miyokoshimura

Top programming languages on GitHub

RANKED BY COUNT OF DISTINCT USERS CONTRIBUTING TO PROJECTS OF EACH LANGUAGE.



What do all these languages have in common?

Non-turing complete examples?

What is the difference between a language, framework and module?