Introduction to Information Systems

– Understanding the digital world

5 Algorithms, Programming and Programming languages

Liang Zhao

ILA, Doshisha University

12001102, Fall, 2024

Today's schedule

- Answers to the coding tasks (15')
- Mini test #5 (10')
- Algorithms & Programming (20')
- Programming languages (45')

Last coding task 1

This program calculates 1 + 2 + … + N for a number N given by the user.

Toy Machine Simulator ->

http://www.cs.princeton.edu/courses/archive/fall18/cos109/toysim.html

```
1.  _GET
2.  _STORE N
3.  L_LOAD N
4.  _IFZERO E
5.  _ADD S
6.  _STORE S
7.  _LOAD N
8.  _SUB 1
9.  _STORE N
10. _GOTO L
11. E_LOAD S
12. _PRINT
13. _STOP
14. N
15. S 0
```

Last coding task 2

Write a program that prints 1, 2, …, 99 in that order with loop: a smart answer by students.

Toy Machine Simulator ->

http://www.cs.princeton.edu/courses/archive/fall18/cos109/toysim.html

1. L␣load a
2. ␣add 1
3. ␣store a
4. ␣print
5. ␣sub 99
6. ␣ifzero b
7. ␣goto L
8. b stop
9. a 0

Algorithm is the abstract procedure to solve a task

The efficiency of a program depends on the algorithm.

```
GET
STORE A
ADD A
...
ADD A
PRINT
STOP
A
```

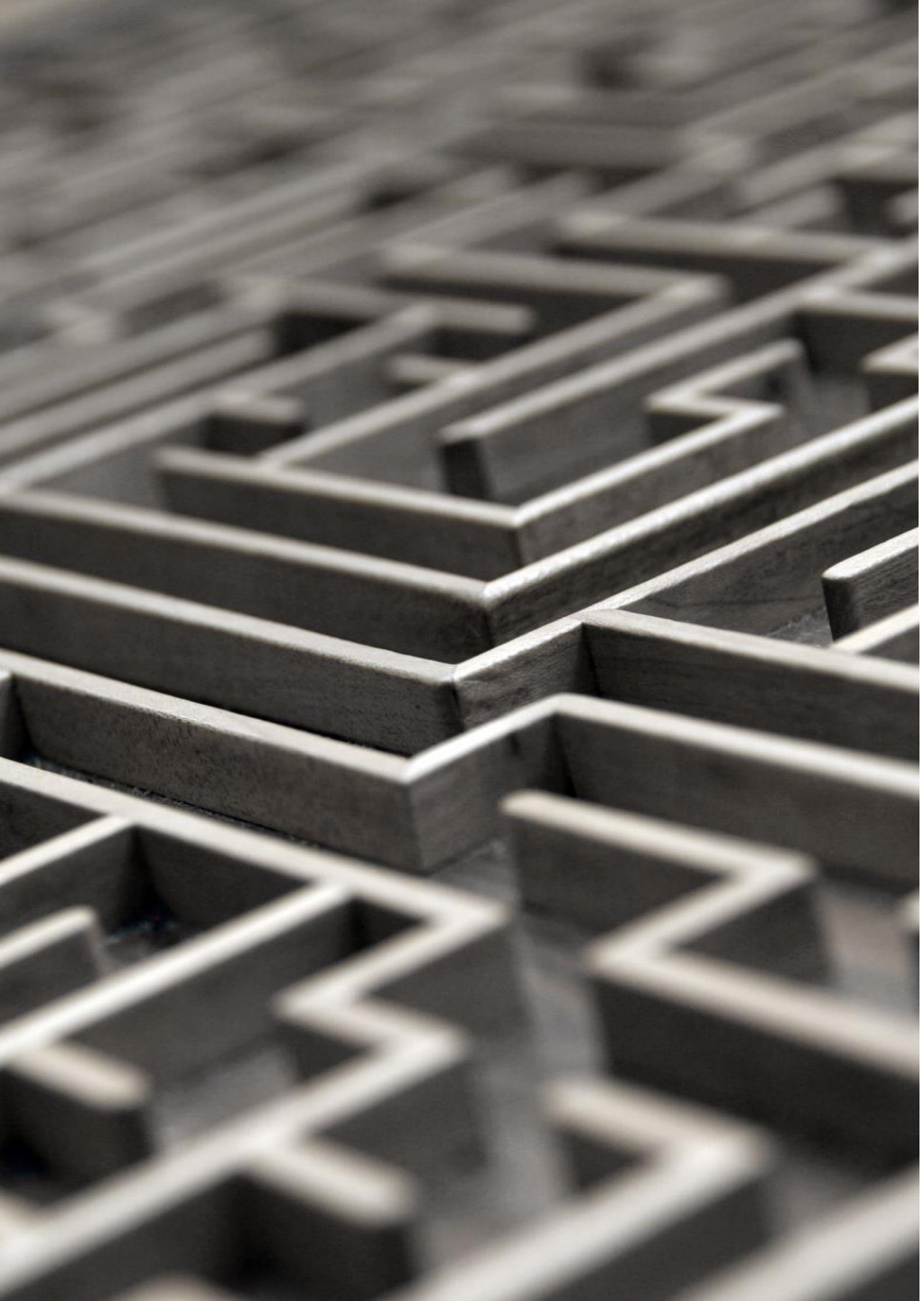
99 times

> 100 operations

```
1.  GET          9.  STORE S
2.  STORE A      10. GOTO L
3.  L LOAD N     11. E LOAD S
4.  IFZERO E     12. PRINT
5.  SUB 1        13. STOP
6.  STORE N      14. A
7.  LOAD S       15. S 0
8.  ADD A        16. N 100
```

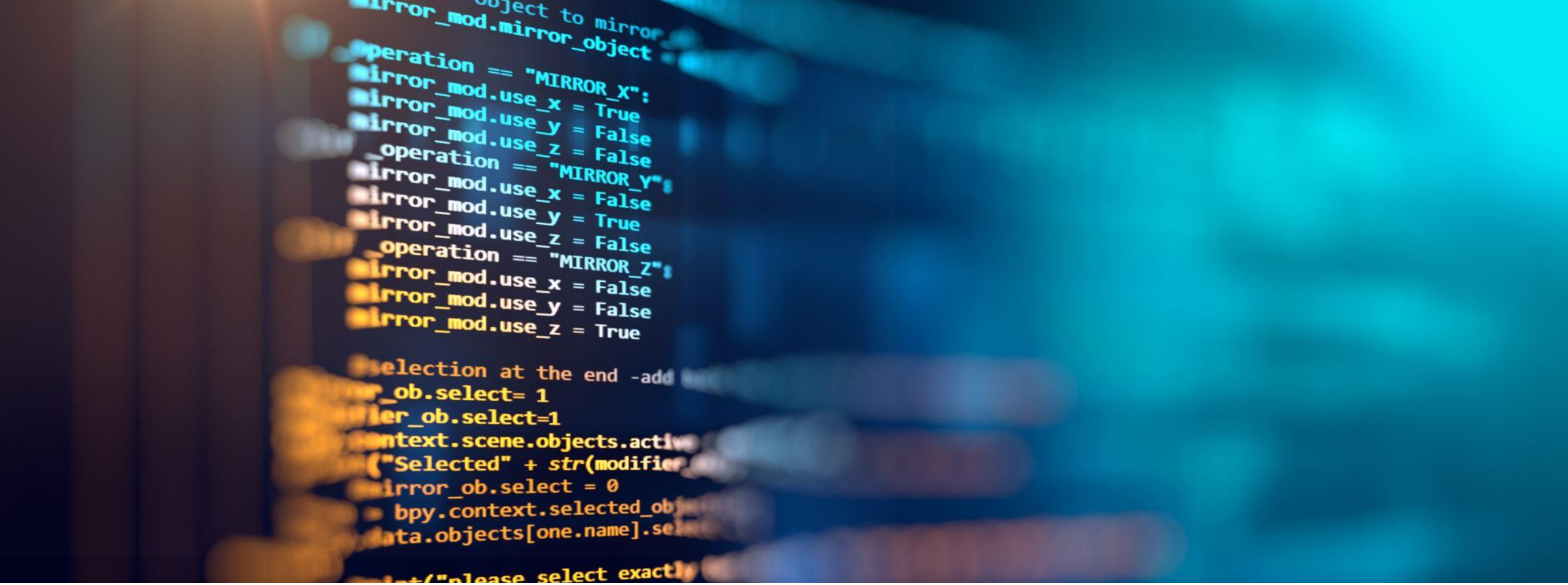
Only 17 operations

```
1.  GET          10. ADD A
2.  STORE A      11. STORE A
3.  ADD A        12. ADD A
4.  STORE A      13. ADD A
5.  ADD A        14. ADD A
6.  STORE A      15. ADD A
7.  ADD A        16. PRINT
8.  ADD A        17. STOP
9.  ADD A        18. A
```



Hard problems and complexity

- **Complexity** is how difficult an algorithm is. It is evaluated by the **#steps** and **#memory** used.
- https://www.youtube.com/watch?v=Q4gTV4rOzRs (See HOW HARD a hard problem can be. It is in Japanese with English subtitles. About 8 minutes)

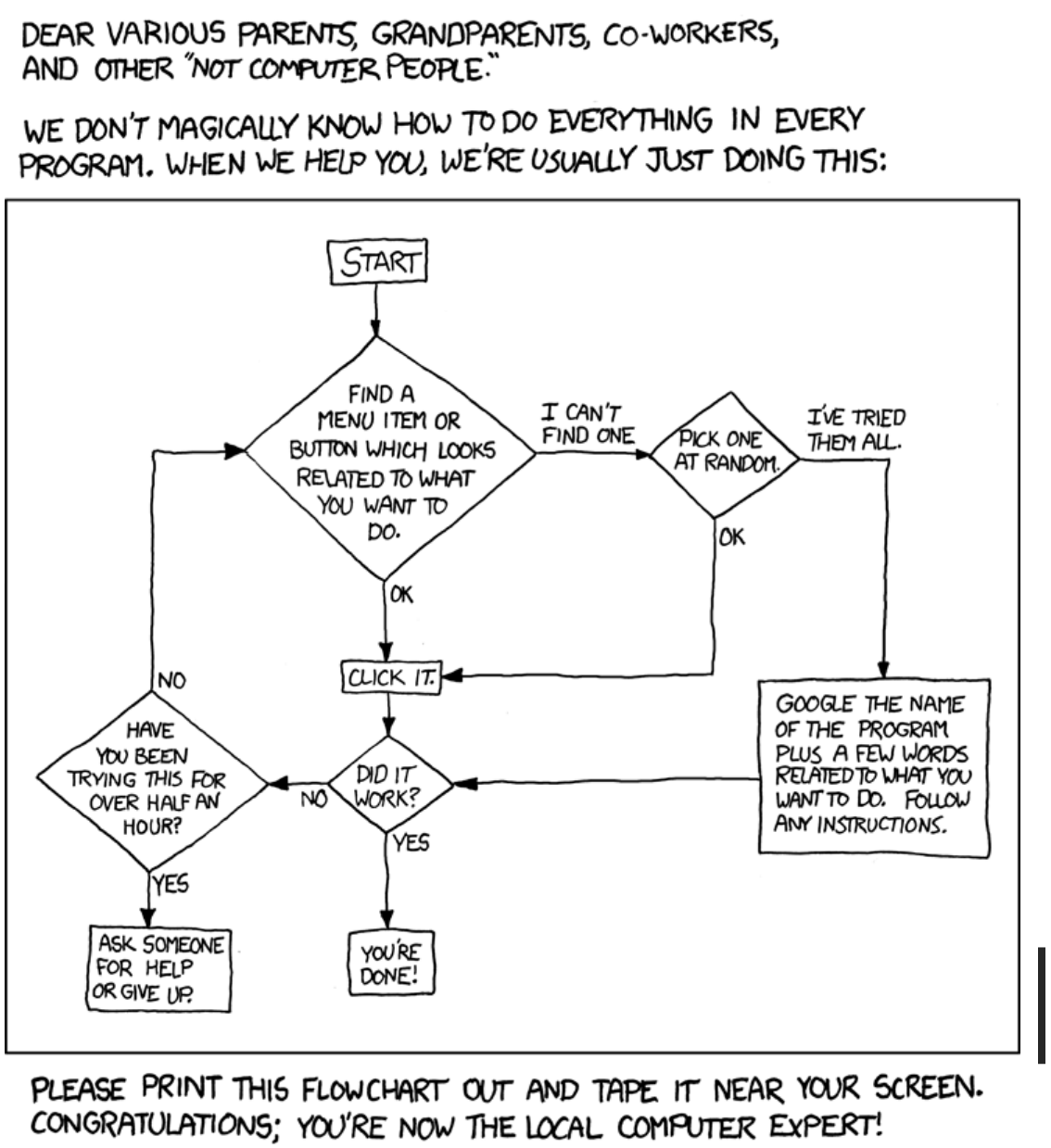


Program

- A collection of instructions to implement an algorithm.
- There exist many dialects, called programming languages.
- The one we learned is assembly. There are > 200 languages.

Three fundamental structures of ALL programs (algorithms).

1. Sequential
2. Conditional
3. Iterative (-> loop)



Python: a high-level programming language

1. A Python program to calculate 1 + 2 + … + N for a number N input by the user.
2. A Python program to print 1, 2, …, 99 in that order with loop.

Program 1

```
1. n = int(input())
2. s = 0
3. for i in range(1, n+1):
4. ____s = s + i
5. print(s)
```

Note: use 4 spaces for the indent.

Program 2

```
1. for i in range(100):
2. ____print(i)
```

https://paiza.io/en/projects/new?language=python3

1. Put the code into the code window (skip # lines).
2. Put the input in the “Input” window.
3. Click the "Run (Ctrl+Enter)" button to execute.

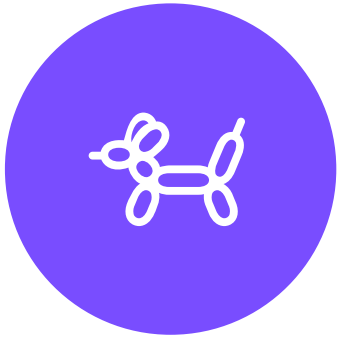
Scratch (yet another programming language)

1. Go to https://scratch.mit.edu
2. “Start Creating“ -> “Tutorials“
3. Open the next sample (click flag -> start, click cat -> stop)

https://scratch.mit.edu/projects/750318208/

Click “See inside“ to investigate the code. Modify the program so that the cat runs faster, or slower, or says something different, or does something different, etc. Notice how the three fundamental structures are implemented in Scratch.

Homework



WATCH VIDEOS

Watch the movies mentioned so far if you have not
(You are not expected to understand everything)

READ THE TEXTBOOK IF YOU HAVE NOT