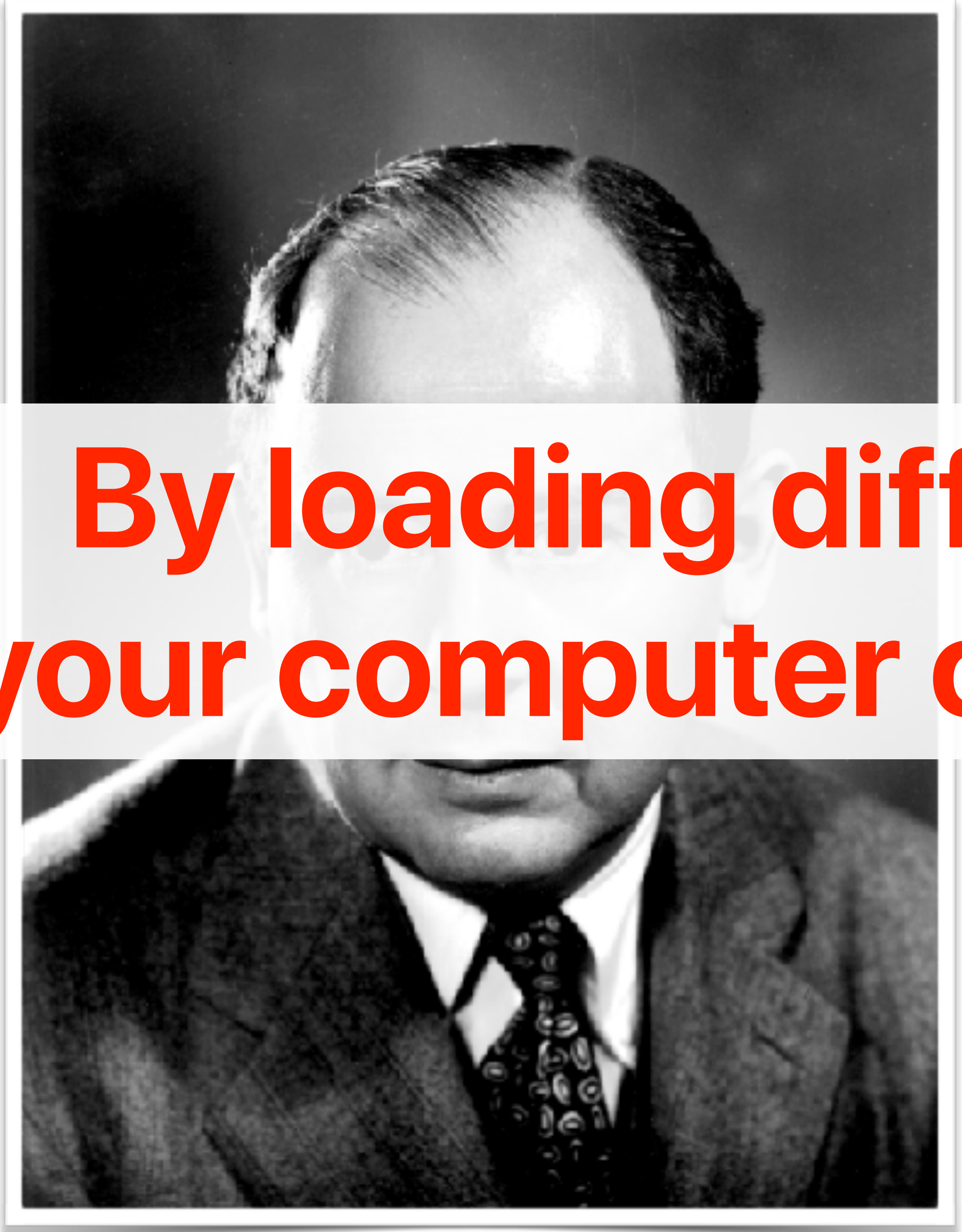
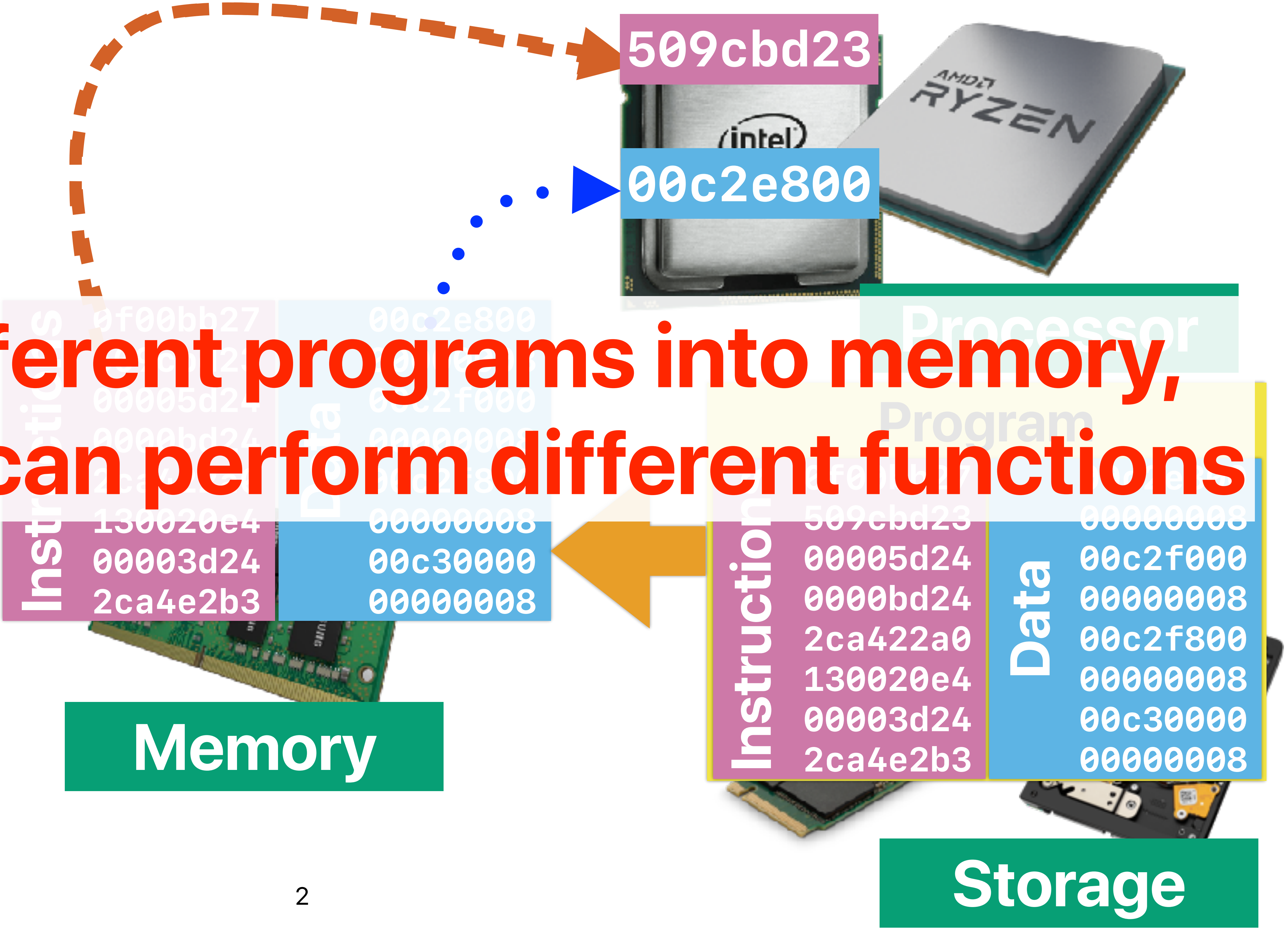
Performance (I)

Hung-Wei Tseng

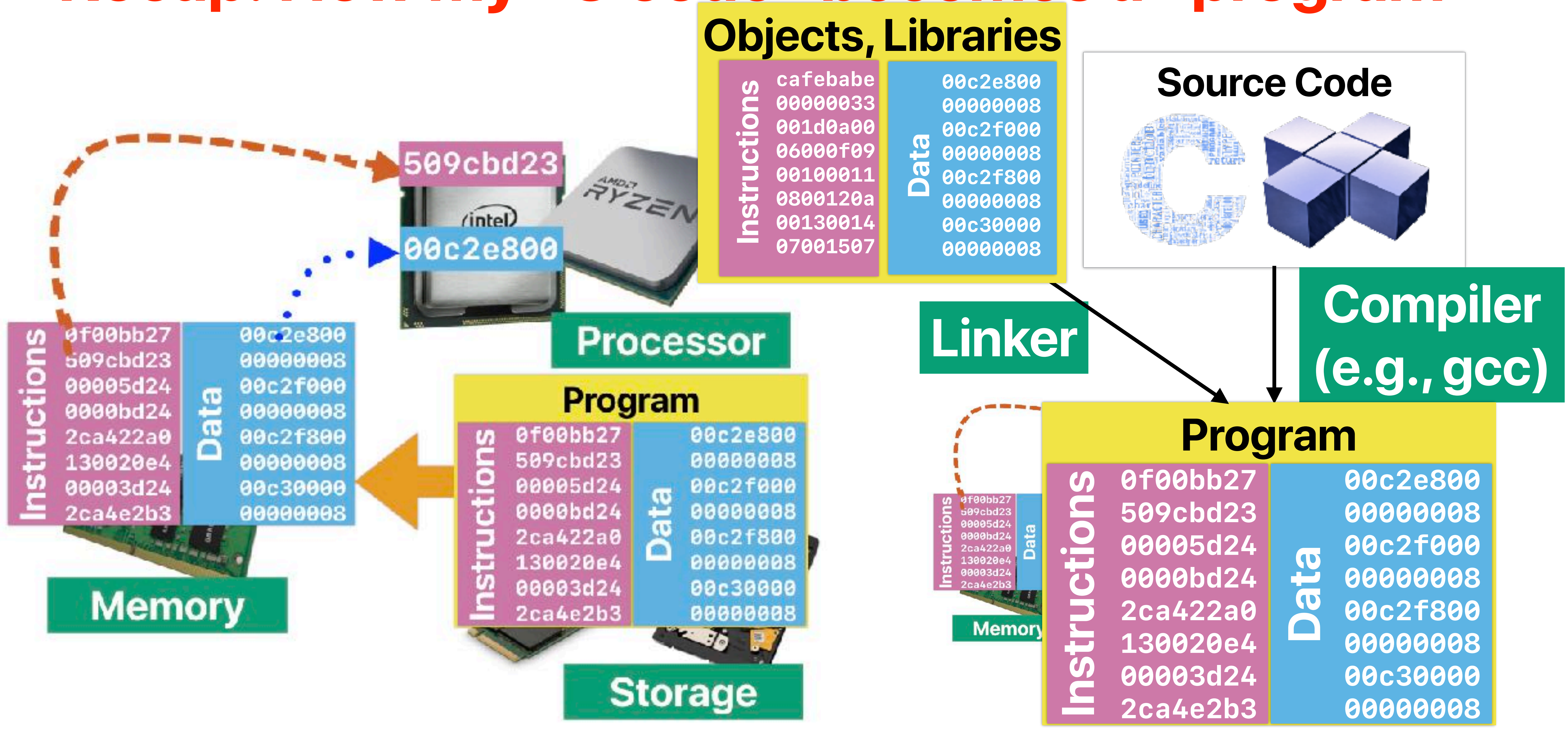
Recap: von Neumman Architecture



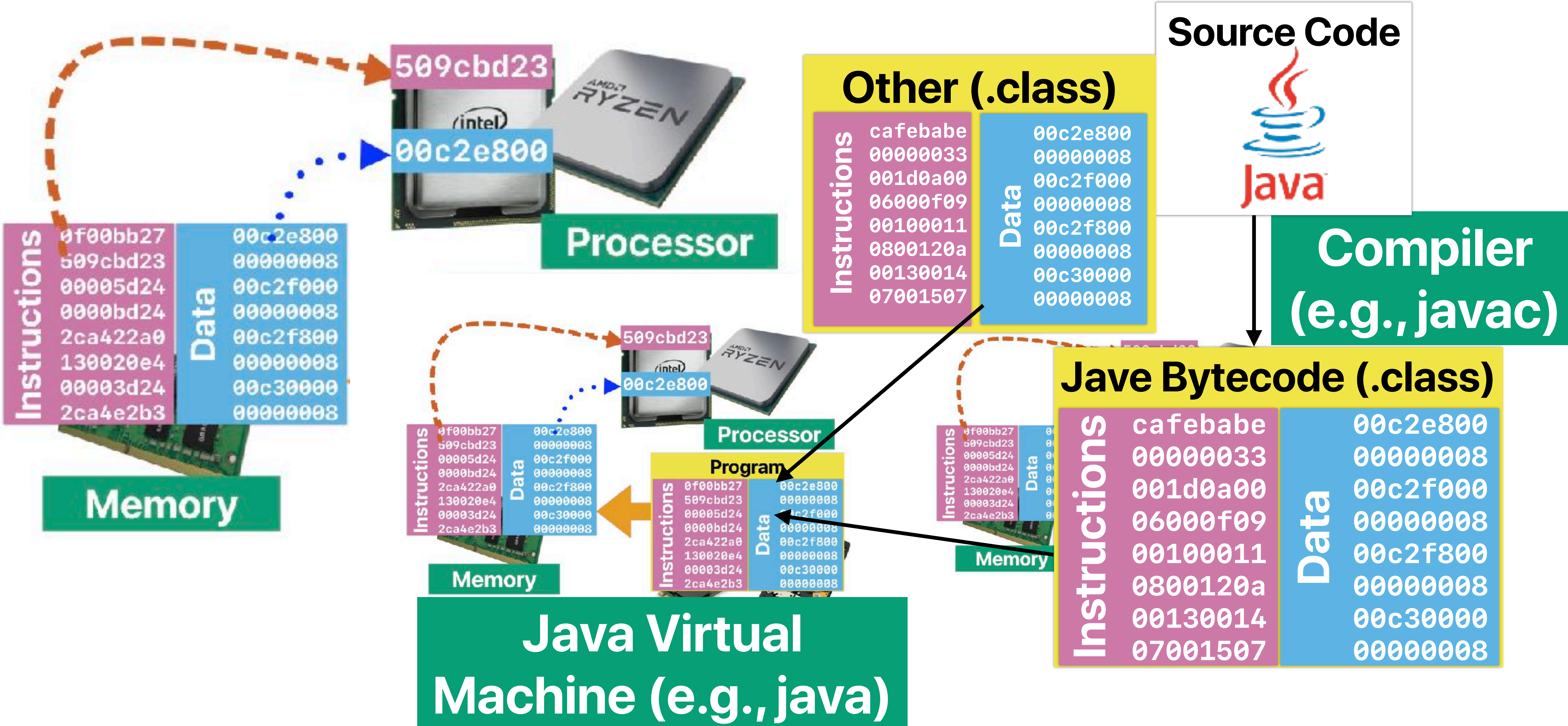


By loading different programs into memory, your computer can perform different functions

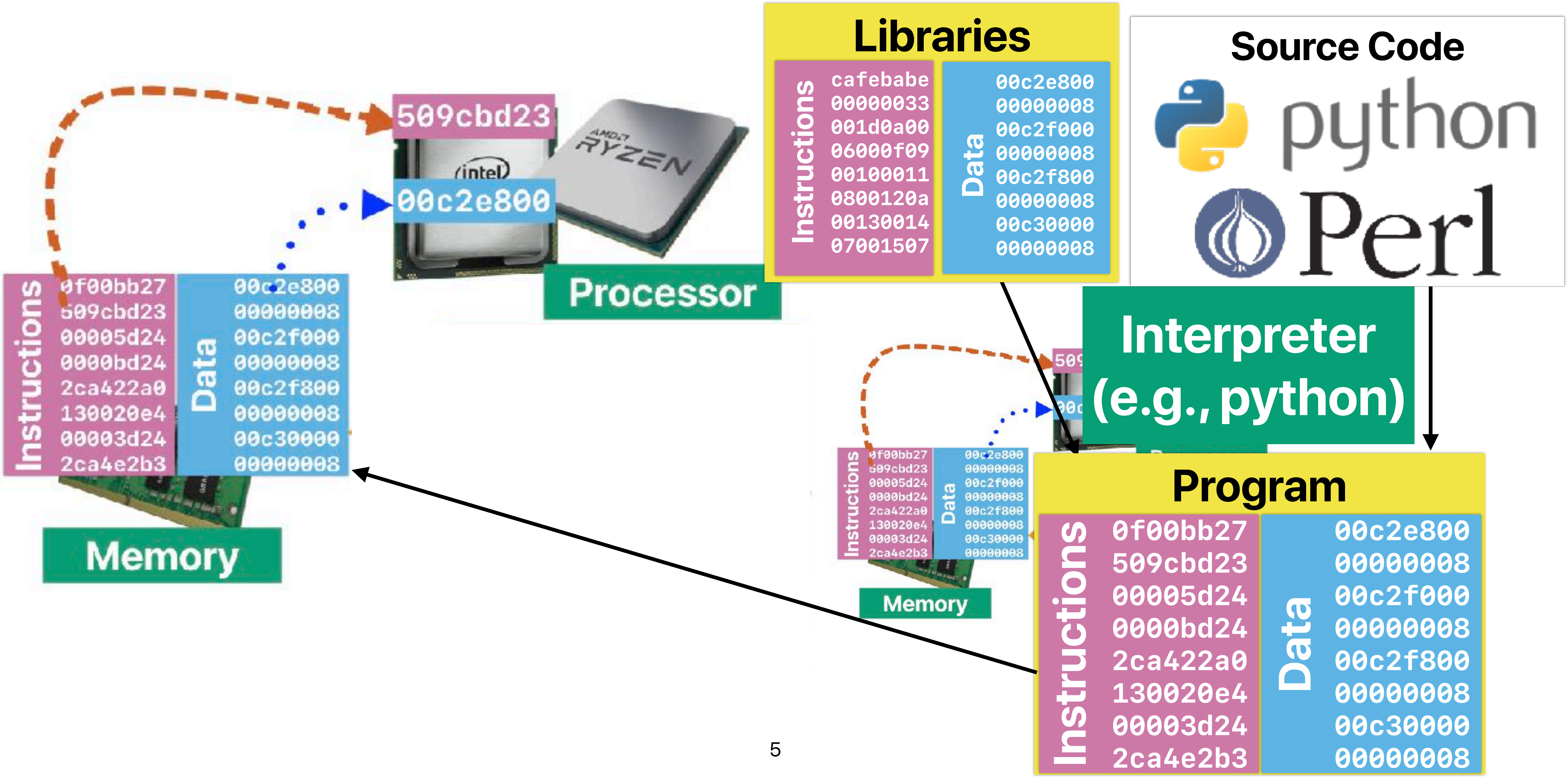
Recap: How my “C code” becomes a “program”



Recap: How my “Java code” becomes a “program”



Recap: How my “Python code” becomes a “program”



Outline

- Definition of “Performance”
- What affects each factor in “Performance Equation”
- Instruction Set Architecture & Performance

Definition of “Performance”

https://www.pollev.com/hungweitseng close in 01:00

CPU Performance Equation (X)

- Assume that we have an application composed with a total of **5000000000** instructions, in which **20%** of them are "Type-A" instructions with an average **CPI of 8** cycles, **20%** of them are "Type-B" instructions with an average **CPI of 4** cycles and **the rest** instructions are "Type-C" instructions with average **CPI of 1** cycle. If the processor runs at **3 GHz**, how long is the execution time?
    A. 3.67 sec
    B. 5 sec
    C. 6.67 sec
    D. 15 sec
    E. 45 sec

CPU Performance Equation

$$Performance = \frac{1}{Execution\ Time}$$

$$Execution\ Time = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

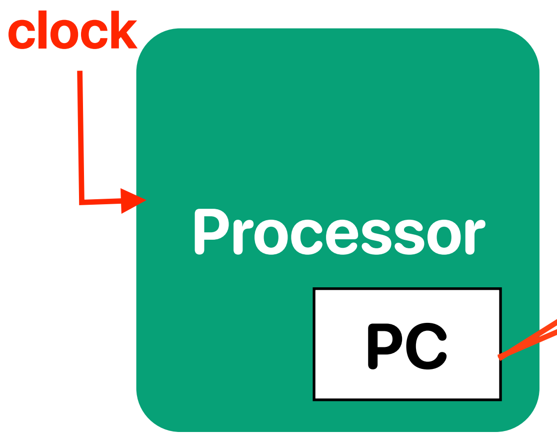
$$ET = IC \times CPI \times CT$$

$1GHz = 10^9 Hz = \frac{1}{10^9} sec\ per\ cycle = 1\ ns\ per\ cycle$

$$\frac{1}{Frequency(i.e., clock\ rate)}$$

Execution Time

- The simplest kind of performance
- Shorter execution time means better performance
- Usually measured in seconds



instruction memory

```
120007a30:   0f00bb27   ldah   gp,15(t12)
120007a34:   509cbd23   lda    gp,-25520(gp)
120007a38:   00005d24   ldah   t1,0(gp)
120007a3c:   0000bd24   ldah   t4,0(gp)
120007a40:   2ca422a0   ldl    t0,-23508(t1)
120007a44:   130020e4   beq    t0,120007a94
120007a48:   00003d24   ldah   t0,0(gp)
120007a4c:   2ca4e2b3   stl    zero,-23508(t1)
120007a50:   0004ff47   clr    v0
120007a54:   28a4e5b3   stl    zero,-23512(t4)
120007a58:   20a421a4   ldq    t0,-23520(t0)
120007a5c:   0e0020e4   beq    t0,120007a98
120007a60:   0204e147   mov    t0,t1
120007a64:   0304ff47   clr    t2
120007a68:   0500e0c3   br     120007a80
```

How many of these?

How long is it take to execution each of these?

$$\frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

Performance Equation (X)

- Assume that we have an application composed with a total of **5000000000** instructions, in which **20%** of them are "Type-A" instructions with an average **CPI of 8** cycles, **20%** of them are "Type-B" instructions with an average **CPI of 4** cycles and the rest instructions are "Type-C" instructions with average **CPI of 1** cycle. If the processor runs at **3 GHz**, how long is the execution time?
  A. 3.67 sec
  B. 5 sec
  C. 6.67 sec
  D. 15 sec
  E. 45 sec

$$ET = (5 \times 10^9) \times \underbrace{(20\% \times 8 + 20\% \times 4 + 60\% \times 1)}_{\text{average CPI}} \times \frac{1}{3 \times 10^{-9}} sec = 5$$

average CPI

$$ET = IC \times CPI \times CT$$

https://www.pollev.com/hungweitseng close in 01:00

Speedup of Y over X

- Consider the same program on the following two machines, X and Y. By how much Y is faster than X?

	Clock Rate	Instructions	Percentage of Type-A	CPI of Type-A	Percentage of Type-B	CPI of Type-B	Percentage of Type-C	CPI of Type-C
Machine X	3 GHz	500000	20%	8	20%	4	60%	1
Machine Y	5 GHz	500000	20%	13	20%	4	60%	1

A. 0.2
B. 0.25
C. 0.8
D. 1.25
E. No changes

Speedup

- The relative performance between two machines, X and Y. Y is n times faster than X

$$n = \frac{Execution\ Time_X}{Execution\ Time_Y}$$

- The speedup of Y over X

$$Speedup = \frac{Execution\ Time_X}{Execution\ Time_Y}$$

Speedup of Y over X

- Consider the same program on the following two machines, X and Y. By how much Y is faster than X?

	Clock Rate	Instructions	Percentage of Type-A Insts.	CPI of Type-A Insts.	Percentage of Type-B Insts.	CPI of Type-B Insts.	Percentage of Type-C Insts.	CPI of Type-C Insts.
Machine X	3 GHz	500000	20%	8	20%	4	60%	1
Machine Y	5 GHz	500000	20%	13	20%	4	60%	1

$$ET_Y = (5 \times 10^9) \times (20\% \times 13 + 20\% \times 4 + 60\% \times 1) \times \frac{1}{5 \times 10^{-9}} sec = 4$$

$$Speedup = \frac{Execution\ Time_X}{Execution\ Time_Y}$$

$$= \frac{5}{4} = 1.25$$

A. 0.2
B. 0.25
C. 0.8
D. 1.25
E. No changes

What Affects Each Factor in Performance Equation

https://www.pollev.com/hungweitseng close in 01:00

How programmer affects performance?

- Performance equation consists of the following three factors
  ① IC
  ② CPI
  ③ CT

How many can a **programmer** affect?

A. 0
B. 1
C. 2
D. 3

Programmer's impact

- By adding the "sort" in the following code snippet, what the programmer changes in the performance equation to achieve **better** performance?

```
std::sort(data, data + arraySize);

for (unsigned c = 0; c < arraySize*1000; ++c) {
        if (data[c%arraySize] >= INT_MAX/2)
            sum ++;
    }
}
```

A. CPI
B. IC
C. CT
D. IC & CPI

Programmer's impact

- By adding the "sort" in the following code snippet, what the programmer changes in the performance equation to achieve **better** performance?

```
std::sort(data, data + arraySize);

for (unsigned c = 0; c < arraySize*1000; ++c) {
        if (data[c%arraySize] >= INT_MAX/2)
            sum ++;
    }
}
```

A. CPI

B. IC — we increased IC, suppose to make the performance worse

C. CT

D. IC & CPI

https://www.pollev.com/hungweitseng close in 01:00

Demo — programmer & performance

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
  for(j = 0; j < ARRAY_SIZE; j++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
  for(i = 0; i < ARRAY_SIZE; i++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

- How many of the following make(s) the performance different between version A & version B?
  - ① IC
  - ② CPI
  - ③ CT
  - A. 0
  - B. 1
  - C. 2
  - D. 3

Demo — programmer & performance

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
  for(j = 0; j < ARRAY_SIZE; j++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
  for(i = 0; i < ARRAY_SIZE; i++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

A		B
$O(n^2)$	Complexity	$O(n^2)$
Same	Instruction Count?	Same
Same	Clock Rate	Same
???	CPI	???

Use "performance counters" to figure out!

- Modern processors provides performance counters
  - instruction counts
  - cache accesses/misses
  - branch instructions/mis-predictions
- How to get their values?
  - You may use "perf stat" in linux
  - You may use Instruments —> Time Profiler on a Mac
  - Intel's vtune — only works on Windows w/ intel processors
  - You can also create your own functions to obtain counter values

Demo — programmer & performance

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
  for(j = 0; j < ARRAY_SIZE; j++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
  for(i = 0; i < ARRAY_SIZE; i++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

A		B
$O(n^2)$	Complexity	$O(n^2)$
Same	Instruction Count?	Same
Same	Clock Rate	Same
Better	CPI	Worse

Demo — programmer & performance

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
  for(j = 0; j < ARRAY_SIZE; j++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
  for(i = 0; i < ARRAY_SIZE; i++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

- How many of the following make(s) the performance different between version A & version B?
  ① IC
  ② CPI ✓
  ③ CT
  A. 0
  B. 1
  C. 2
  D. 3

Programmers can also set the cycle time

https://software.intel.com/sites/default/files/comment/1716807/how-to-change-frequency-on-linux-pub.txt

```
================================================
Subject: setting CPU speed on running linux system


If the OS is Linux, you can manually control the CPU speed by reading and writing some virtual files in the "/proc"

1.) Is the system capable of software CPU speed control?
If the "directory" /sys/devices/system/cpu/cpu0/cpufreq exists, speed is controllable.
-- If it does not exist, you may need to go to the BIOS and turn on EIST and any other C and P state control and vi



2.) What speed is the box set to now?
Do the following:
$ cd /sys/devices/system/cpu
$ cat ./cpu0/cpufreq/cpuinfo_max_freq
3193000
$ cat ./cpu0/cpufreq/cpuinfo_min_freq
1596000

3.) What speeds can I set to?
Do
$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
It will list highest settable to lowest; example from my NHM "Smackover" DX58SO HEDT board, I see:
3193000 3192000 3059000 2926000 2793000 2660000 2527000 2394000 2261000 2128000 1995000 1862000 1729000 159600
You can choose from among those numbers to set the "high water" mark and "low water" mark for speed.  If you set "h


4.) Show me how to set all to highest settable speed!
Use the following little sh/ksh/bash script:
$ cd /sys/devices/system/cpu  # a virtual directory made visible by device drivers
$ newSpeedTop=`awk '{print $1}' ./cpu0/cpufreq/scaling_available_frequencies`
$ newSpeedLow=$newSpeedTop  # make them the same in this example
$ for c in ./cpu[0-9]* ; do
>   echo $newSpeedTop >${c}/cpufreq/scaling_max_freq
>   echo $newSpeedLow >${c}/cpufreq/scaling_min_freq
> done
$

5.) How do I return to the default - i.e. allow machine to vary from highest to lowest?
Edit line # 3 of the script above, and re-run it.  Change the line:
$ newSpeedLow=$newSpeedTop  # make them the same in this example
To read
```

How programmer affects performance?

- Performance equation consists of the following three factors
  ① IC
  ② CPI
  ③ CT

  How many can a **programmer** affect?

  A. 0
  B. 1
  C. 2
  D. 3

https://www.pollev.com/hungweitseng close in 01:00

How programming languages affect performance

- Performance equation consists of the following three factors
  1. IC
  2. CPI
  3. CT

How many can the **programming language** affect?

A. 0
B. 1
C. 2
D. 3

https://www.pollev.com/hungweitseng close in 01:00

Programming languages

- Which of the following programming language needs to highest instruction count to print "Hello, world!" on screen?
  - A. C
  - B. C++
  - C. Java
  - D. Perl
  - E. Python

Programming languages

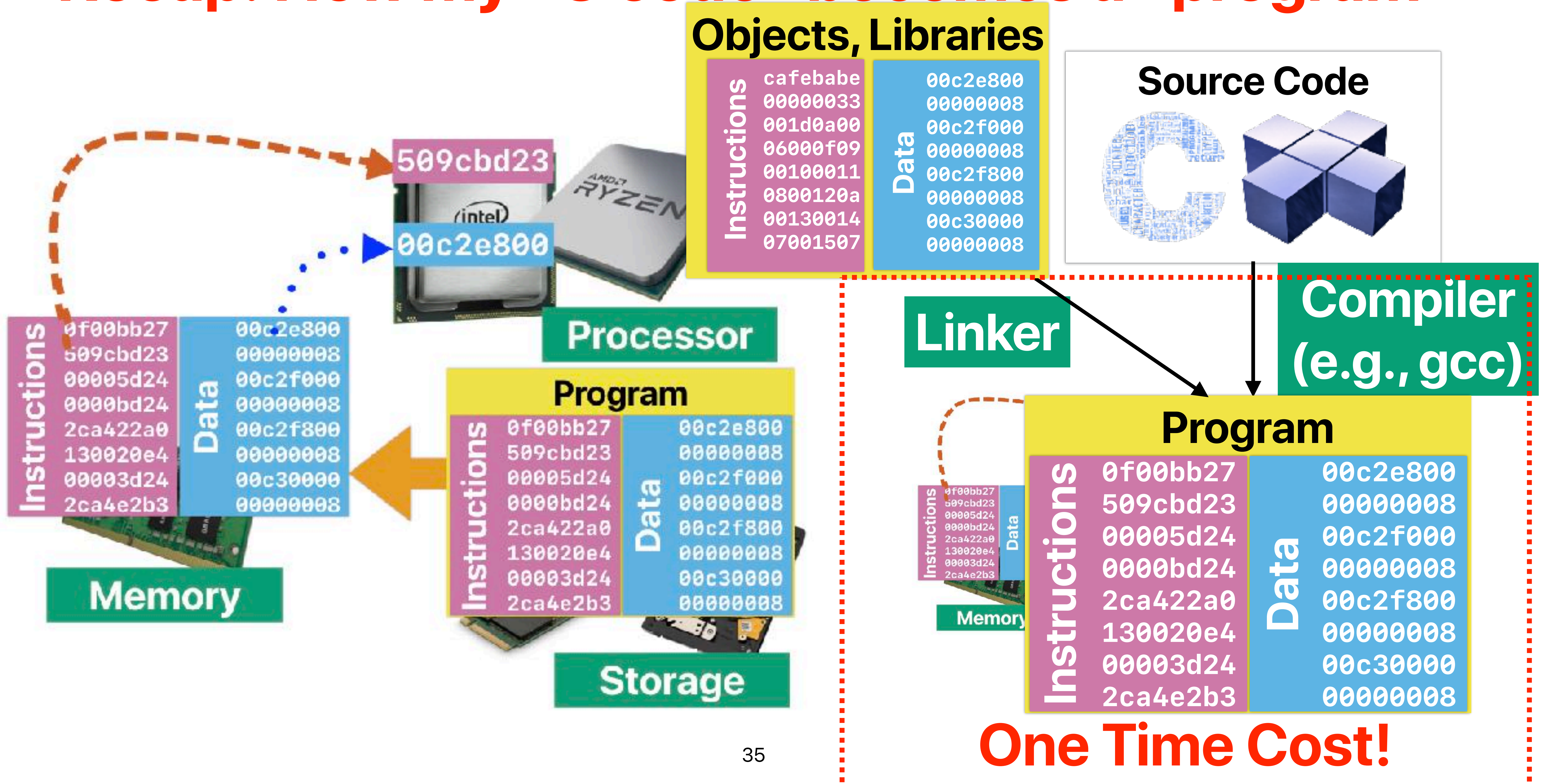
- How many instructions are there in “Hello, world!”

	Instruction count	LOC	Ranking
C	600k	6	1
C++	3M	6	2
Java	~210M	8	5
Perl	10M	4	3
Python	~30M	1	4

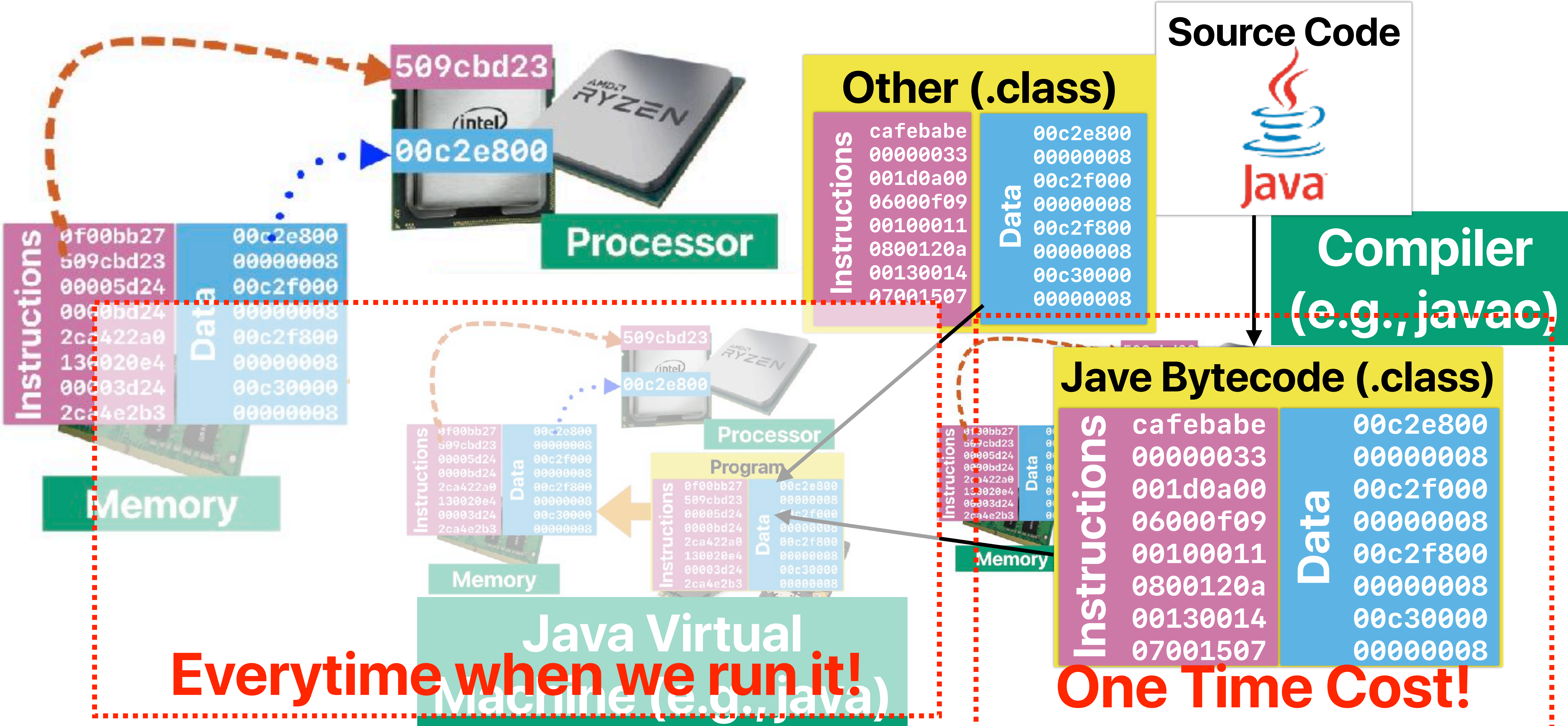
Programming languages

- Which of the following programming language needs to highest instruction count to print "Hello, world!" on screen?
    A. C
    B. C++
    C. Java
    D. Perl
    E. Python

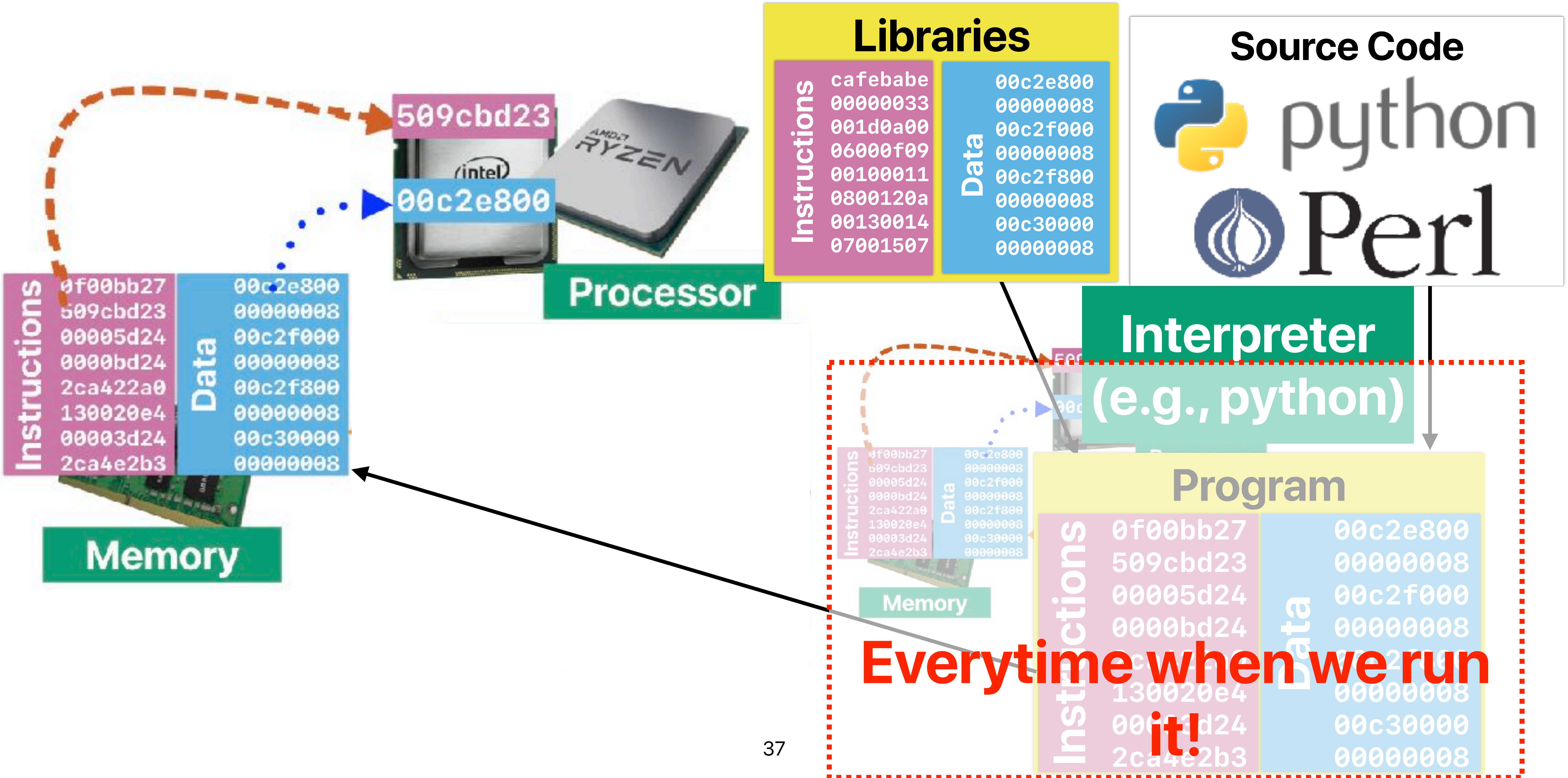
Recap: How my "C code" becomes a "program"



Recap: How my "Java code" becomes a "program"



Recap: How my "Python code" becomes a "program"



How programming languages affect performance

- Performance equation consists of the following three factors
  ① IC
  ② CPI
  ③ CT

How many can the **programming language** affect?

A. 0
B. 1
C. 2
D. 3

https://www.pollev.com/hungweitseng close in 01:00

How compilers affect performance

- Performance equation consists of the following three factors
  1. IC
  2. CPI
  3. CT

How many can the **compiler** affect?

A. 0
B. 1
C. 2
D. 3

How compilers affect performance

- Performance equation consists of the following three factors
  ① IC
  ② CPI
  ③ CT

  How many can the **compiler** affect?

  A. 0
  B. 1
  C. 2
  D. 3

Revisited the demo with compiler optimizations!

- gcc has different optimization levels.
  - -O0 — no optimizations
  - -O3 — typically the best-performing optimization

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
  for(j = 0; j < ARRAY_SIZE; j++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
  for(i = 0; i < ARRAY_SIZE; i++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

Demo revisited — compiler optimization

- Compiler can reduce the instruction count, change CPI — with "limited scope"
- Compiler CANNOT help improving "crummy" source code

```
if(option)
    std::sort(data, data + arraySize);
```

Compiler can never add this — only the programmer can!

```
for (unsigned c = 0; c < arraySize*1000; ++c) {
        if (data[c%arraySize] >= INT_MAX/2)
            sum ++;
    }
}
```

How about “computational complexity”

- Algorithm complexity provides a good estimate on the performance if —
  - Every instruction takes exactly the same amount of time
  - Every operation takes exactly the same amount of instructions

These are unlikely to be true

Summary of CPU Performance Equation

$$Performance = \frac{1}{Execution\ Time}$$

$$Execution\ Time = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

$$ET = IC \times CPI \times CT$$

- IC (Instruction Count)
  - ISA, Compiler, algorithm, programming language, **programmer**
- CPI (Cycles Per Instruction)
  - Machine Implementation, microarchitecture, compiler, application, algorithm, programming language, **programmer**
- Cycle Time (Seconds Per Cycle)
  - Process Technology, microarchitecture, **programmer**

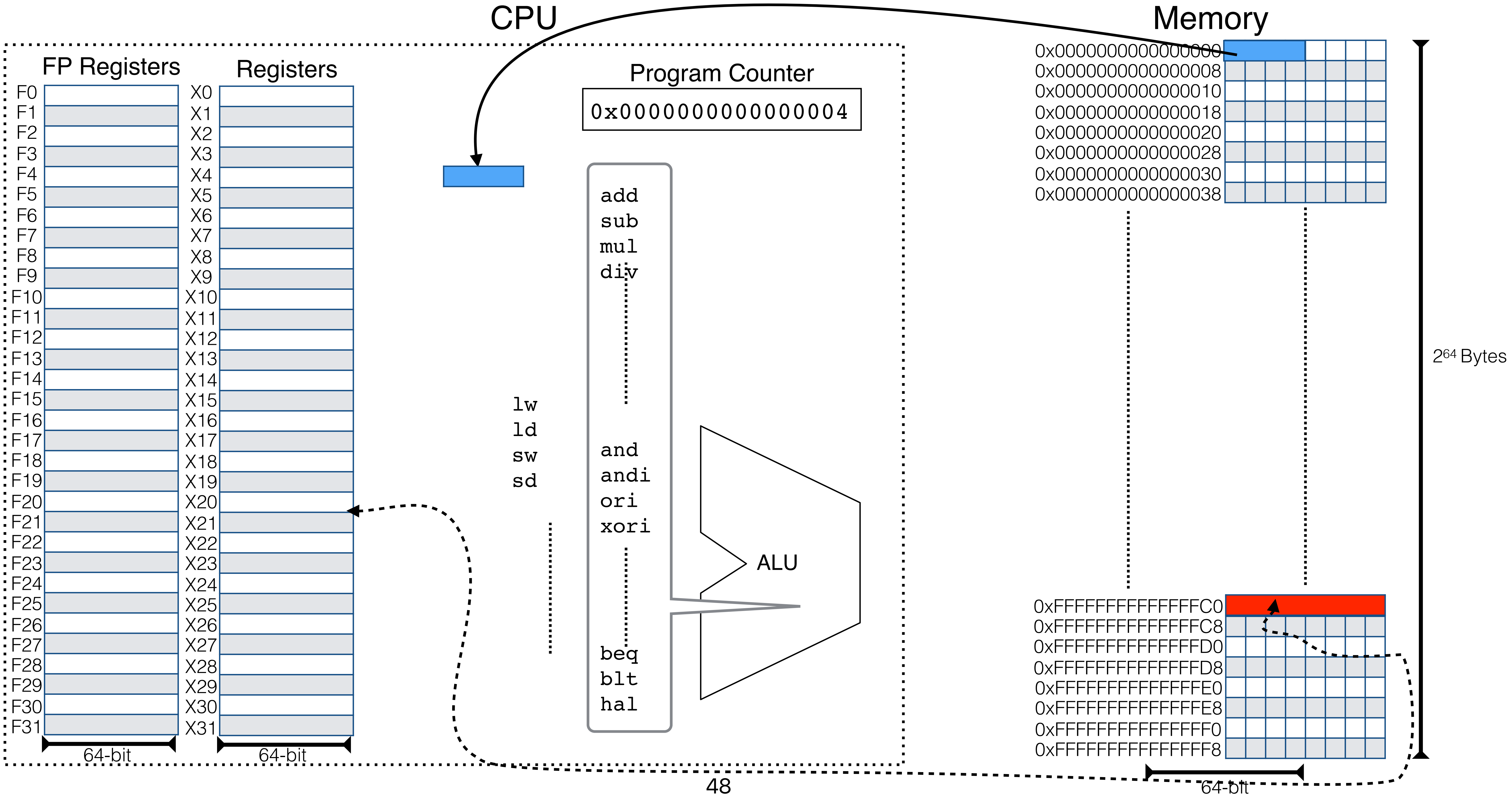
Instruction Set Architecture (ISA) & Performance

Recap: ISA — the interface b/w processor/software

- Operations
  - Arithmetic/Logical, memory access, control-flow (e.g., branch, function calls)
  - Operands
    - Types of operands — register, constant, memory addresses
    - Sizes of operands — byte, 16-bit, 32-bit, 64-bit
- Memory space
  - The size of memory that programs can use
  - The addressing of each memory locations
  - The modes to represent those addresses

Popular ISAs

The abstracted "RISC-V" machine



Announcement

- Office hour of Hung-Wei Tseng changes — **MF 1p-2p @ WCH 406**
- **Josep Torrellas** (ACM, AAAS, IEEE fellow. IEEE TCCA Chair) is giving a talk at UCR!
  - next Monday 11am (Bourns A265)
  - if you attend, please submit a short summary — count as a bonus assignment
- Check our website for slides, iLearn for quizs/assignments, piazza for discussions