Introduction to Computational Thinking and Python Programming

Sarom Leang, Ph.D. (Instructor)

Jesse McClandish (Mentor)

December 3, 2022

Session 2

Names and Faces and Pronouns

Sarom Leang, Ph.D. (Instructor)

- Professor
- Instructor
- Mr. Leang

Jesse McClandish (Mentor)

- Jesse

Schedule

10:00 AM – 10:15 AM **Homeroom**

10:15 AM – 11:35 AM **G1 Block**

11:35 AM – 12:35 PM **Break/Lunch**

12:40 PM – 02:00 PM **G2 Block**

Course Overview

This course introduces the fundamental building blocks of computational thinking and computer programming using the Python language.

Upon successful completion of this course, students will be able to:

- Improve their problem-solving skills
- Write, read, and execute Python code using basic data types and operators

Course Overview (cont.)

Session #1 – October 29, 2022

- ~~Entrance survey~~
- Python programming environment, “Hello World”

Session #2 – December 3, 2022

- Parallel computing
- Python programming environment, “Hello World”
- Python arithmetic operators, integer and float data types

Session #3 – March 4, 2023

- Python Booleans data type, comparison and logical operators, truth tables

Session #4 – March 25, 2023

- Python string data type, control structures, and collections {list, dictionary, tuple}

Session #5 – April 22, 2023

- Decomposition, pattern recognition, abstraction, and algorithms
- Exit survey

Student Expectations

- **NO FOOD**
- **NO DRINKS** (on the table)
- Be respectful to individuals and property
- Be open to learning
- Be open to not understanding
- Be patient with yourself
- Ask questions
- Explore
- **Embrace failure**

Introductions

Choose one:

- Everyone has a story. What is yours?

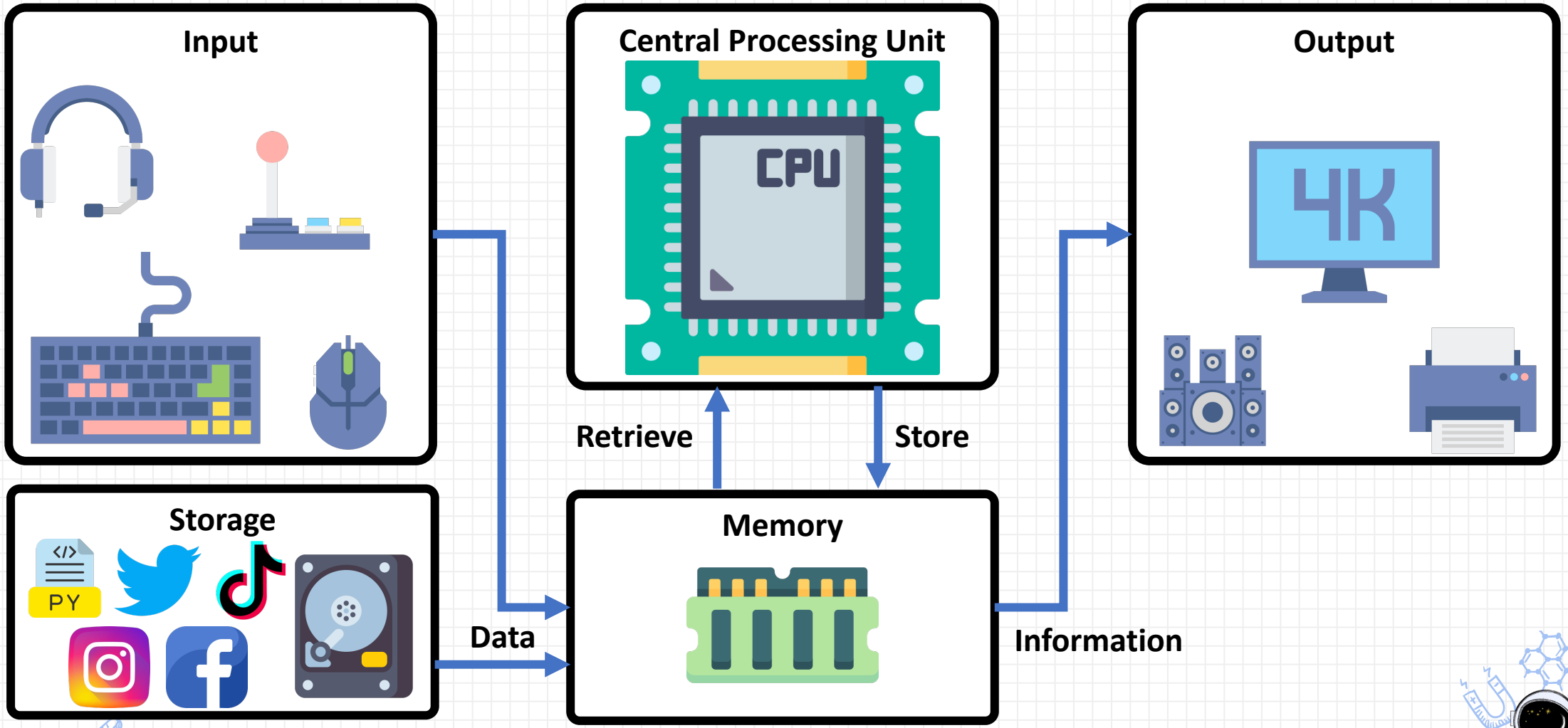
or

- Answer the following questions:
  - Name/Pronouns
  - Grade level and school
  - What did you choose to attend STEM Fusion!?
  - Why did you choose this course for STEM Fusion!?
  - Do you have any experience in computer programming?
  - What do you hope to get out of this course?

My Story

1982	Coming to America!
	Khmer Rouge Genocide
1998	EIP Class 7 Scholar
	Wakefield High School
2004	B.S. Chemistry (Honors, High Distinction), Minor CS
	George Mason University
2011	Ph.D. Physical Chemistry / Postdoctoral Researcher
	Iowa State University
2014	Assistant Research Scientist
	The Ames Laboratory, Department of Energy
2018	Senior Computational Scientist/Senior Software Engineer
	EP Analytics, Inc.

Discussion – How do computers work?



Discussion – What is a computer?

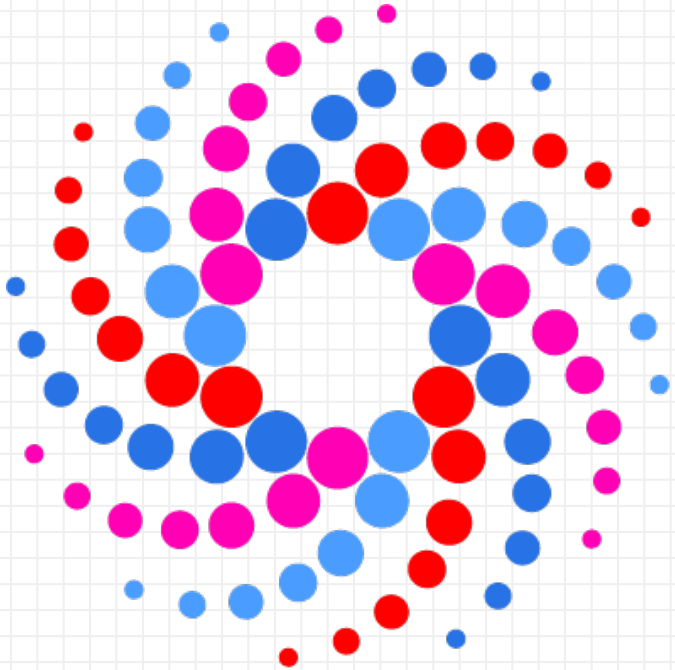
- An electronic device that **stores**, **retrieves**, and **processes** data.
- A **programmable** electronic device that **stores**, **retrieves**, and **processes** data following a **set of instructions**.

program

Discussion – What is computer programming?

- The **mental process** of developing a set of instructions for a computer.
  - Design and planning
- What is the **physical process** called?
  - Coding

Supercomputing Conference



SC22

Dallas, TX | hpc accelerates.

HPC = high performance computing

High Performance

Example	General	High	Metrics
Cars	Consumer e.g., Ford Fusion	Performance e.g., Formula-1, Koenigsegg Agera	• Speed • Torque (force) • Horse-power (force x distance)
Computers	Consumer e.g., Alienware Area 51	Performance e.g., HPE Cray Frontier	• Operations per second • Number of CPUs • Aggregate memory

High Performance Computing

Specs	Alienware Area 51	IBM Summit (4,608 nodes) #5
CPU	1x AMD Ryzen 2950X 16-core/32-threads 4.4 GHz	2x IBM Power9 22-core/96-threads 3.07 GHz
GPU	2x NVIDIA RTX 2080 Ti	6x NVIDIA Volta V100
Memory	64 GB	512 GB
Floating point operations per second (FLOPS)	700 GFLOPS	42000 GFLOPS (per node) 42 TFLOPS (per node) **60x** 200000000 GFLOPS (total) 200 PFLOPS (total) **285,714x**

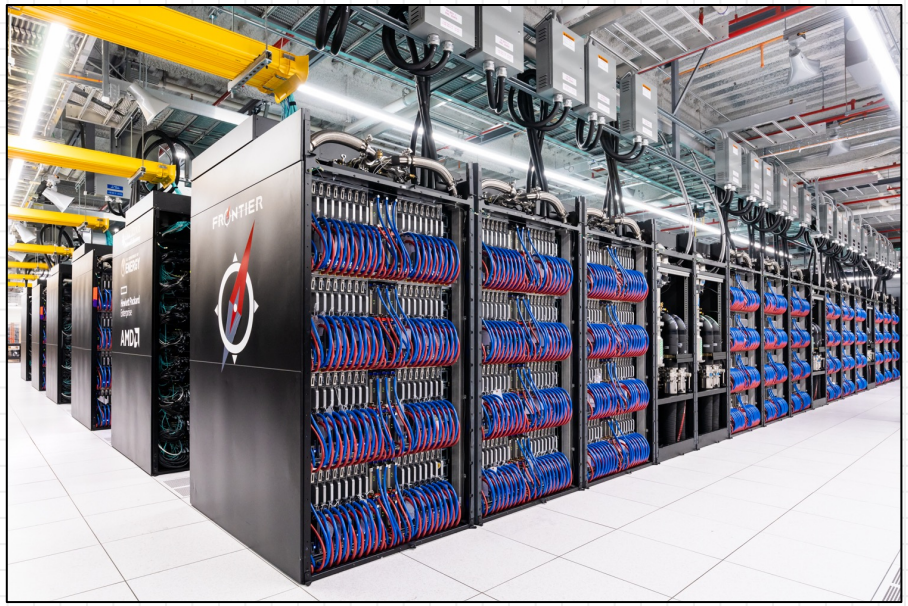
High Performance Computing

Specs	Alienware Area 51	IBM Summit (4,608 nodes) #5
CPU	1x AMD Ryzen 2950X 16-core/32-threads 4.4 GHz	2x IBM Power9 22-core/96-threads 3.07 GHz
GPU	2x NVIDIA RTX 2080 Ti	6x NVIDIA Volta V100
Memory	64 GB	512 GB
Floating point operations per second (FLOPS)	700 GFLOPS	42000 GFLOPS (per node) 42 TFLOPS (per node) **60x** 200000000 GFLOPS (total) 200 PFLOPS (total) **285,714x**

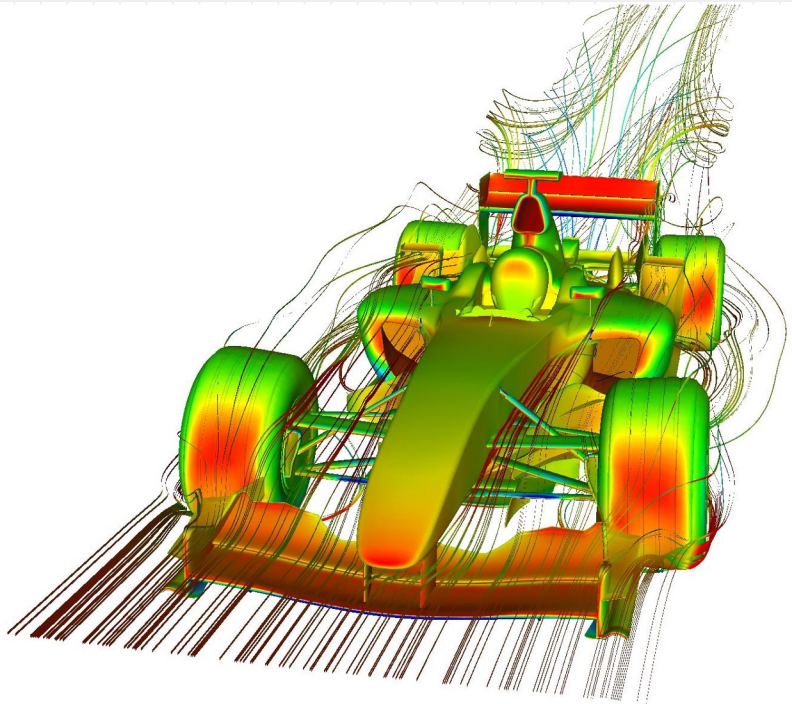
Top 5 Supercomputers

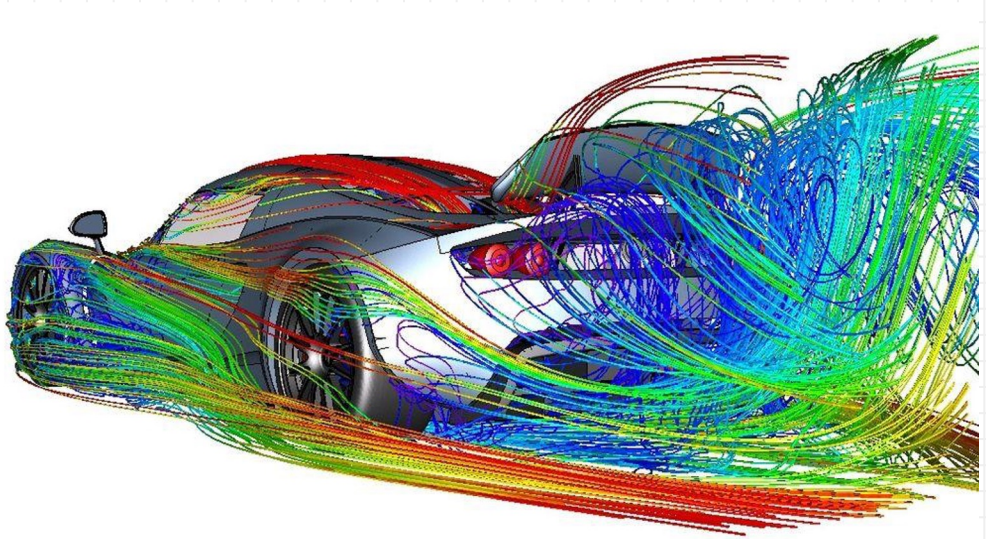
Rank	Country / System	Rpeak (PFlops/s)
1	**USA** / Frontier AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE	1,685
2	**Japan** / Fugaku A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu	537
3	**Finland** / LUMI AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE	428
4	**Italy** / Leonardo Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos	255
5	**USA** / Summit IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM	200

Top 5 Supercomputers



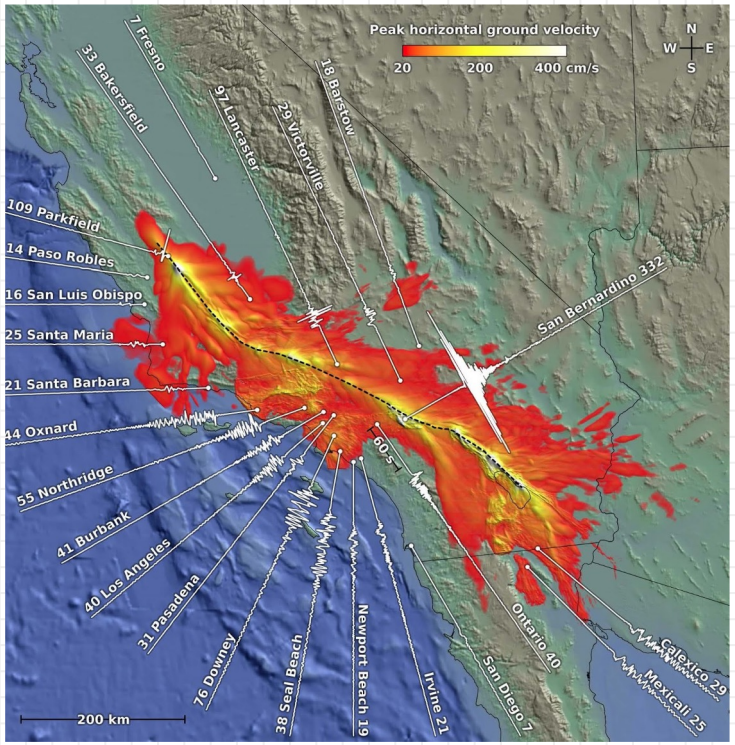
What types of problems are explored with HPC?

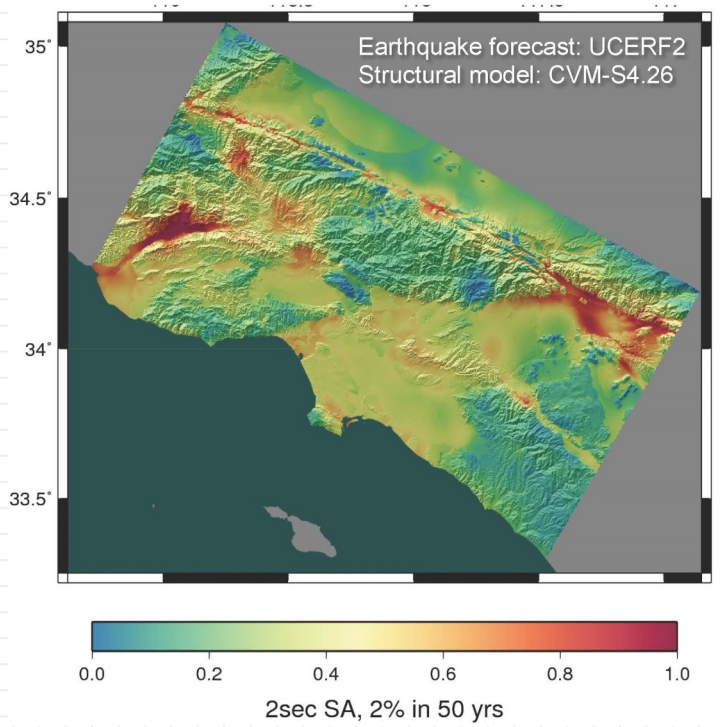




Computational Fluid Dynamics (CFD)

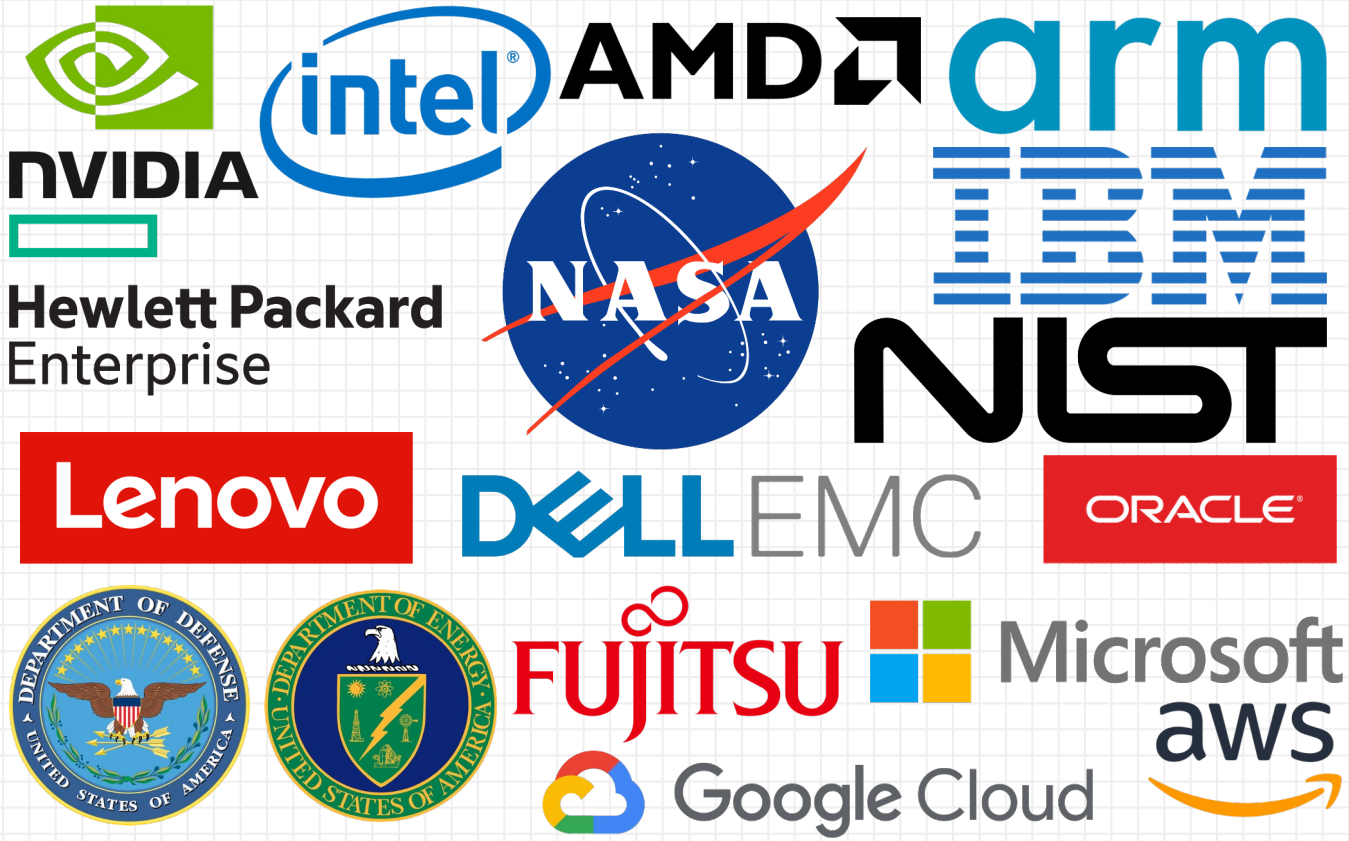
What types of problems are explored with HPC?



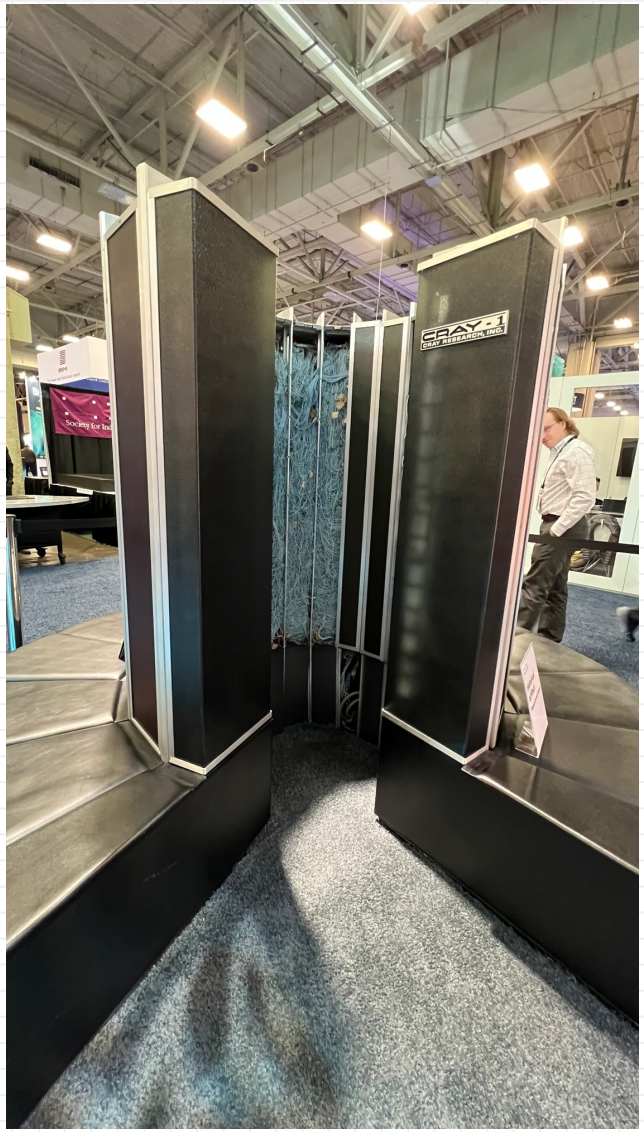


Simulations

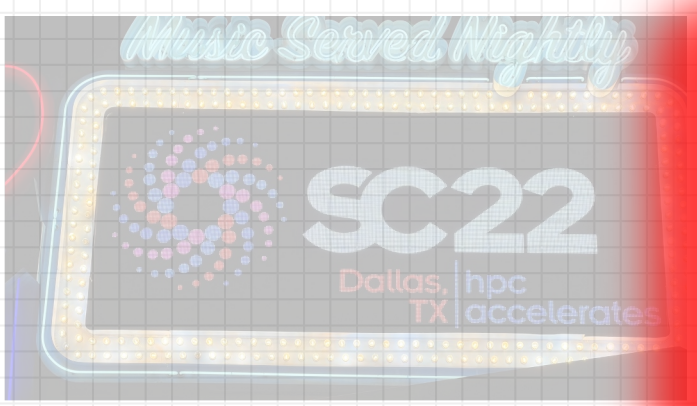
Who is interested in HPC?

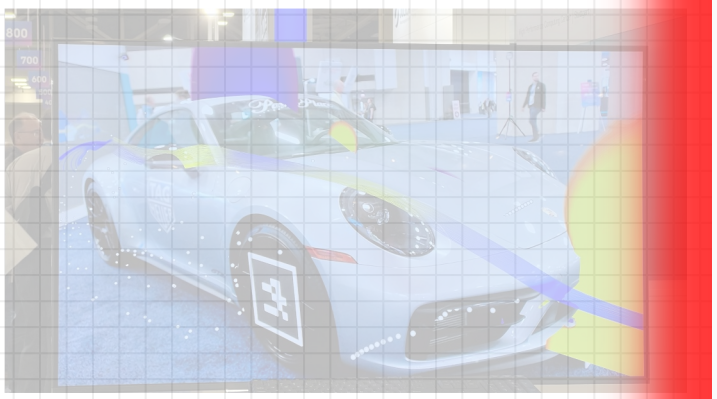


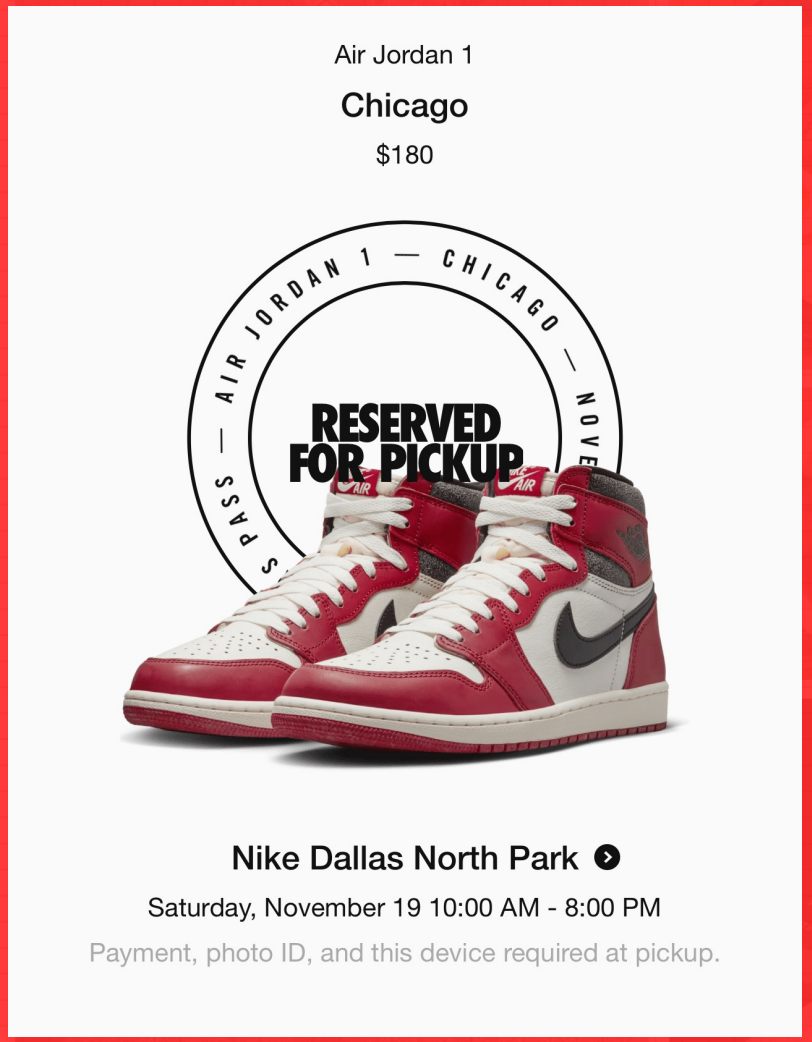
Who is interested in HPC?

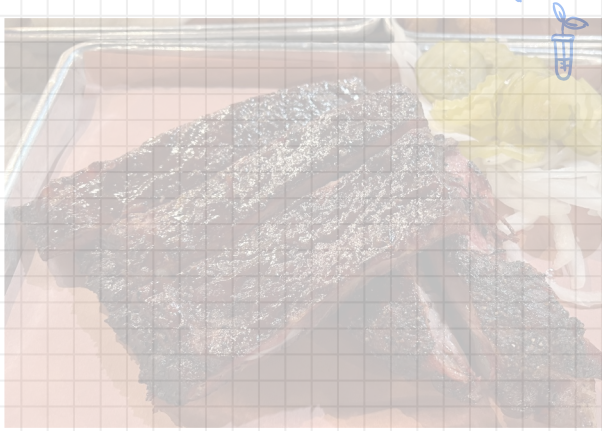


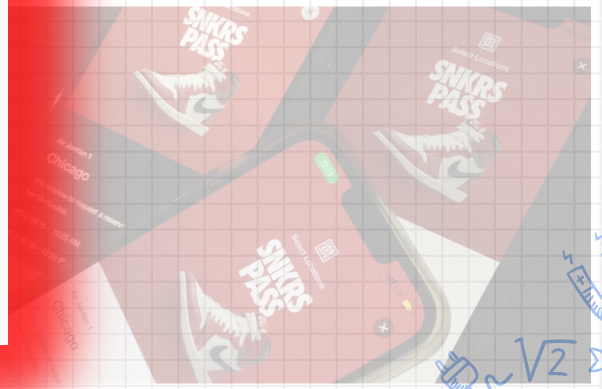
Who is interested in HPC?











Exercise – Human Supercomputer?

- **Challenge**: 300 piece jigsaw puzzle
- **Goal**: Get the most number of correctly connected pieces within the established time frame
- We will assign groups of (or close to):
  - 1
  - 2
  - 4
  - 8
  - 16

Exercise – Human Supercomputer?

- **Challenge**: 300 piece jigsaw puzzle
- **Goal**: Get the most number of correctly connected pieces within the established time frame
- We will assign groups of (or close to):
  - 1
  - 2
  - 4
  - 8
  - 16

# Solvers	G1 (# connected)	G2 (# connected)
1		
2		
4		
8		
16		

Python

- An interpreted high-level programming language released in 1991.
- Can be used for web development, software development, mathematics, and system scripting.
- Works on different platforms (Windows, Mac, Linux, etc.).
- Simple syntax similar to the English language.
  - Syntax – arrangement of words/phrases to create something understandable

Programming Environment

- A text editor
  - Sublime https://www.sublimetext.com
- A Python interpreter
  - Python 3.11.0 https://www.python.org

hello.py

```
print("Hello World!")
```

hello.py

```
print("Hello World!")
```

function

hello.py

```
print("Hello World!")
```

function input

hello.py

```
print("Hello World!")
```

function input

What/Where is the output?

First Programming Exercise: Hello World

- Open up a blank document in the ATOM editor
- Type: `print("Hello World!")`
- Save the file as **hello.py** into your Documents folder
- Open up the command prompt
  1. Test if python is working: type `python` and hit enter
  2. If you get an error then: type
     `set PATH=%PATH%;C:\Python311`
     and hit enter
  3. Repeat step 1
- Execute your script:
  - Switch into the directory containing **hello.py**
    - `cd Documents`
  - Call the python program to interpret and execute your code
    - `python hello.py`

Discussion – What are computers good at?

Discussion – What are computers good at?

- Math (computations)
- Processing power is measured in Hertz (Hz)
  - 1 Hz = 1 clock cycle per second = 1 CPU fetch and execution of an instruction
  - A 1 Hz CPU can do 1 instruction per second
  - 2 GHz = 2 billion clock cycles per second = 2 billion instructions per second!
- NASA Apollo Guidance Computer (1960s) : 0.043 MHz = 0.043×10^6 Hz
- Apple iPhone 14 (2022) : 17 THz = 17×10^{12} Hz

$$\frac{17 \times 10^{12}}{0.043 \times 10^{6}} \approx 4 \times 10^{8} = 400{,}000{,}000 \text{ times faster}$$

Arithmetic Operators in Python

Operator	Usage	Operation
+	x + y	Addition
–	x – y	Subtraction
*	x * y	Multiplication
/	x / y	Division
%	x % y	Remainder
**	x ** y	Exponentiation
//	x // y	Floor division

Second Programming Exercise: Maths!!

Operation	Result
Let x = 8 and y = 2	
`print(x + y)`	
`print(x – y)`	
`print(-x)`	
`print(x*y)`	
`print(x/y)`	
`print(x//y)`	
`print(x%y)`	
`print(x**y)`	

Third Programming Exercise: More Maths!!

Operation	Result
Let x = 9 and y = 4	
`print(x + y)`	
`print(x – y)`	
`print(-x)`	
`print(x*y)`	
`print(x/y)`	
`print(x//y)`	
`print(x%y)`	
`print(x**y)`	

Operator Precedence

- In Python, operators will be evaluated in order of precedence.
- Order of operations – **PEMDAS**
  1. **P**arentheses ( )
  2. **E**xponentiation **
  3. **M**ultiplication * and **D**ivision / , // , %
  4. **A**ddition + and **S**ubtraction –
- After **PEMDAS**, order goes left to right.
- Use parentheses to override order.

Class Exercise: PEMDAS

Try to determine the solution by hand and then verify with Python

```
1 + 2 * 4 = ?
```

```
(1 + 2) * 4 = ?
```

```
5 + (4 - 2) * 2 + 4 % 2 - 4 // 3 - (5 - 3) / 1 = ?
```

Class Exercise: PEMDAS

```
1 + 2 * 4 = ?
1 + 2 * 4 = ?
1 + 8 = 9
```

```
(1 + 2) * 4 = ?
(1 + 2) * 4 = ?
3 * 4 = 12
```

```
5 + (4 - 2) * 2 + 4 % 2 - 4 // 3 - (5 - 3) / 1 = ?
```

Class Exercise: PEMDAS

```
5 + (4 - 2) * 2 + 4 % 2 - 4 // 3 - (5 - 3) / 1 = ?
5 + (4 - 2) * 2 + 4 % 2 - 4 // 3 - (5 - 3) / 1 = ?
5 + 2 * 2 + 4 % 2 - 4 // 3 - 2 / 1 = ?
5 + 2 * 2 + 4 % 2 - 4 // 3 - 2 / 1 = ?
5 + 4 + 0 - 1 - 2 = 6
```