

Intro to Computer Science

Previous

- NULL

Next

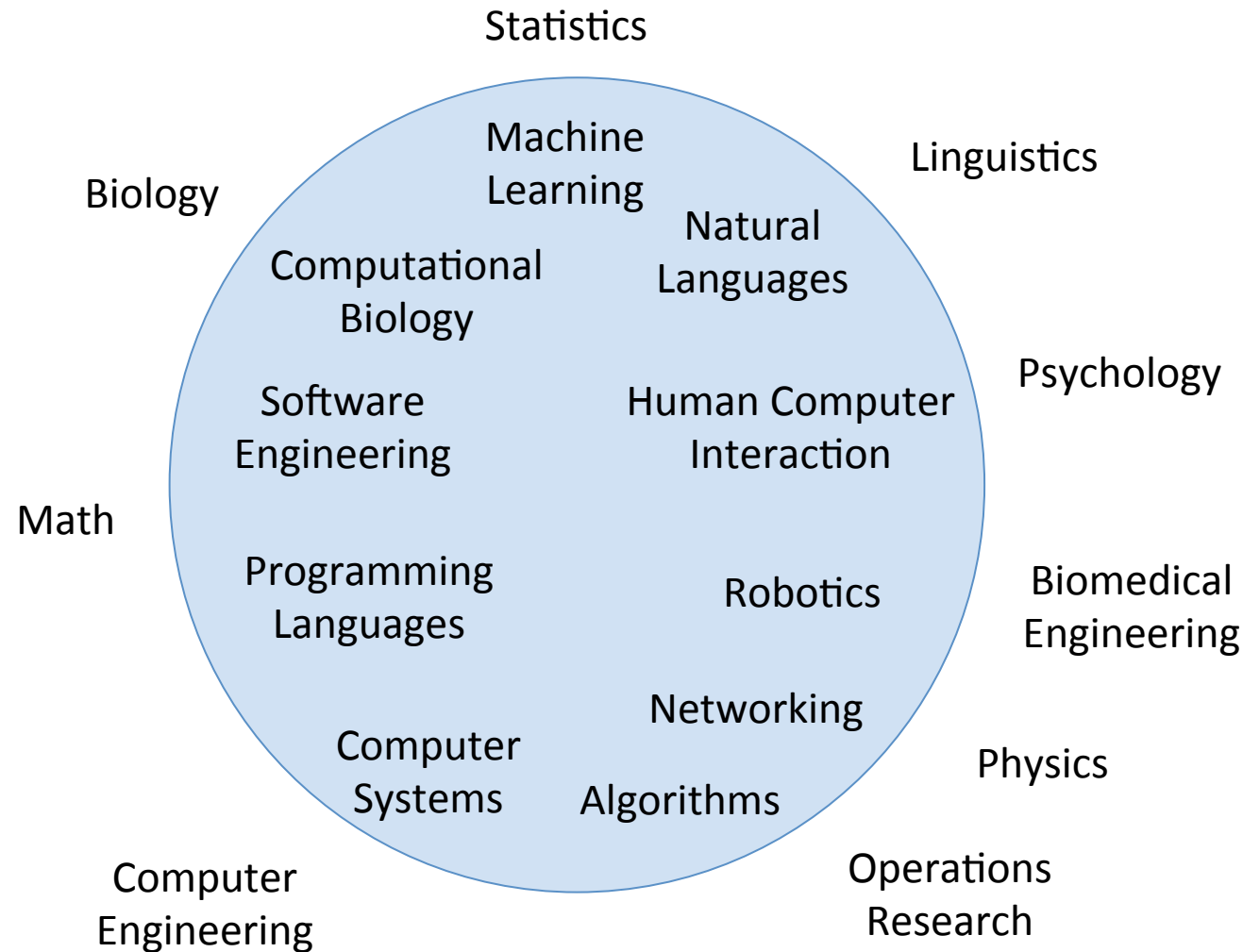
- What is computer science
- Programming languages
- Syntax v. Semantics
- Python!

Readings

Gaddis

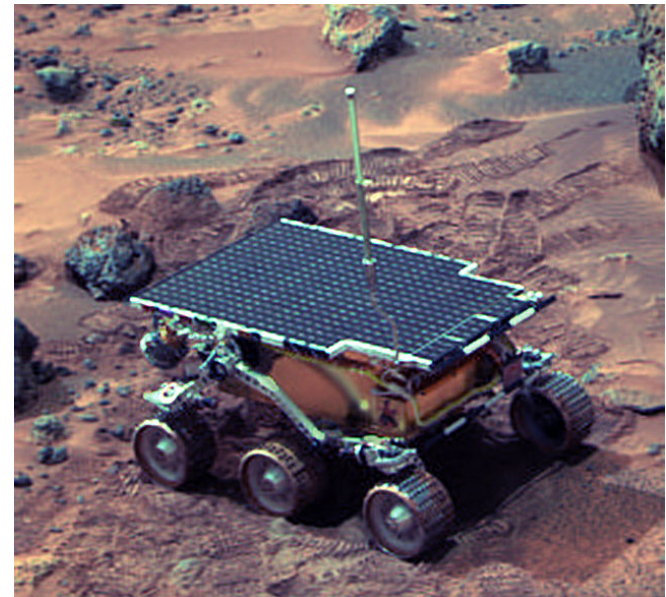
- Chapter 1

Many Facets of Computer Science



The Digital Age

- More and more things around us are driven by computers
- Computers are actually just collections of electronics driven by software
 - Programs are software
- Knowing how to program means you can *contribute* to the digital age



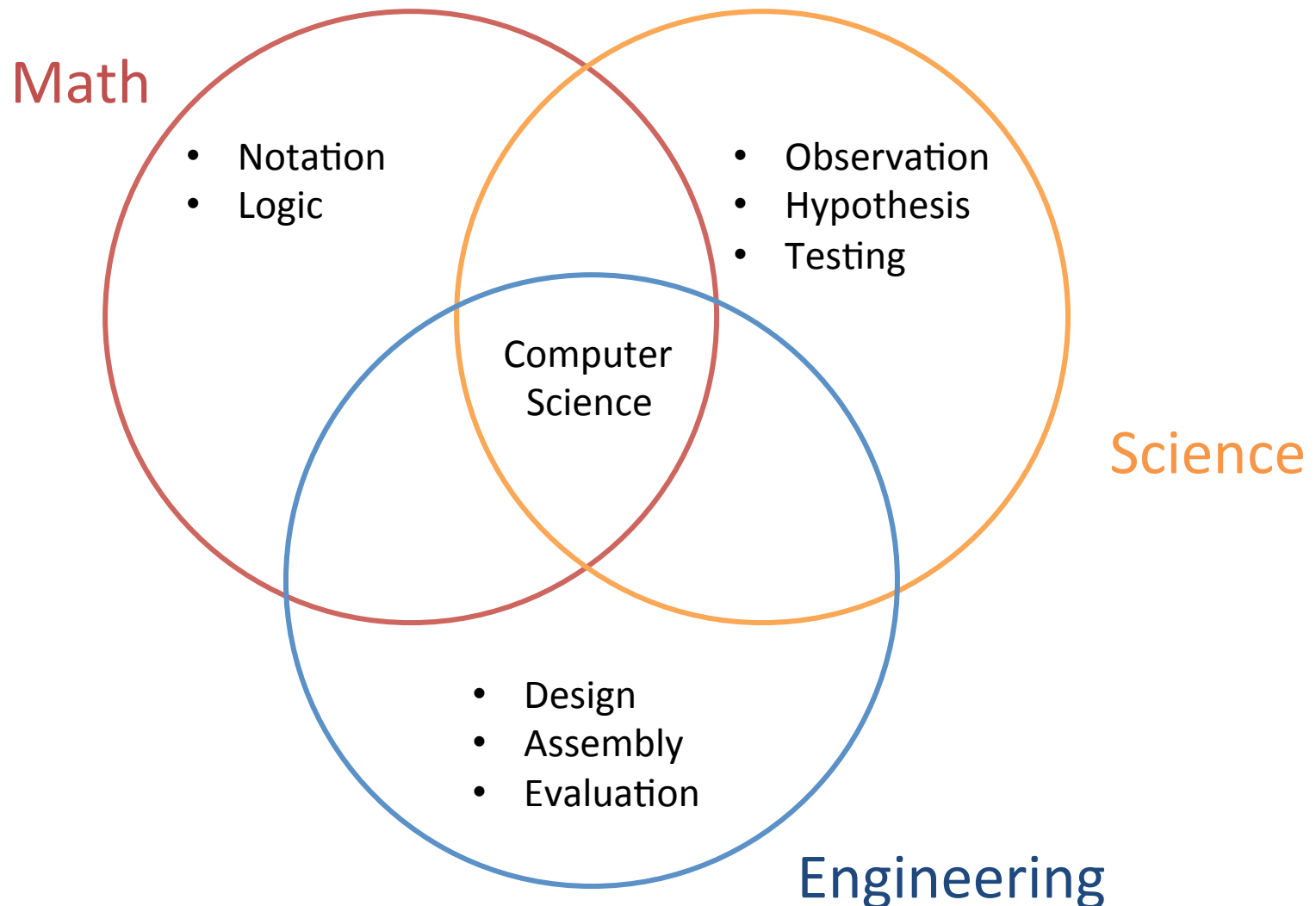
Computers Make Important Decisions



- Computers aren't smart, they just do what humans tell them
 - Humans often get things wrong
- Programming can be an incredible responsibility!



What is Computer Science?



What is Computer Science?

- Formalization of human thinking
- Computational thinking
 - Solving problems through abstraction
 - Creating recipes that implement the solution
 - Using computers to follow the recipes
- This is something you already do!

Concepts: Abstraction

Decomposition

- Reformulate a difficult problem into one we already know how to solve

Abstraction

- Pulling out important details
- Identifying principles that apply to other problems/situations

Concepts: Abstraction

Decomposition

- Reformulate a difficult problem into one we already know how to solve

Abstraction

- Pulling out important details
- Identifying principles that apply to other problems/situations

Which is not like the others?

- | | | |
|---|--|--|
| 1 | People standing in line at a store | |
| 2 | List of jobs waiting to be printed | |
| 3 | Set of tennis balls in their container | |
| 4 | Vehicles lined up at a toll booth | |
| 5 | Patients waiting to see the doctor | |

Concepts: Abstraction

Decomposition

- Reformulate a difficult problem into one we already know how to solve

Abstraction

- Pulling out important details
- Identifying principles that apply to other problems/situations

Which is not like the others?

1	People standing in line at a store	queue
2	List of jobs waiting to be printed	queue
3	Set of tennis balls in their container	stack
4	Vehicles lined up at a toll booth	queue
5	Patients waiting to see the doctor	queue

Concepts: Abstraction

Decomposition

- Reformulate a difficult problem into one we already know how to solve

Abstraction

- Pulling out important details
- Identifying principles that apply to other problems/situations

Which is not like the others?

1	Files and directories on a hard disk	
2	Parents and children in a genealogy chart	
3	Brackets in a tennis tournament	
4	My friends on Facebook	
5	Animal taxonomy	

Concepts: Abstraction

Decomposition

- Reformulate a difficult problem into one we already know how to solve

Abstraction

- Pulling out important details
- Identifying principles that apply to other problems/situations

Which is not like the others?

1	Files and directories on a hard disk	tree
2	Parents and children in a genealogy chart	tree
3	Brackets in a tennis tournament	tree
4	My friends on Facebook	graph
5	Animal taxonomy	tree

Concepts: Logical Thinking

Inductive reasoning

- From specific examples to general principles
- Examples:
 - $8/1=8$; $27/1=27$; $118/1=118$;...
 - Division by 1 equals the number
 - $3+5=8$; $7+9=16$; $27+23=50$;...
 - Odd plus odd is even

Deductive reasoning

Concepts: Logical Thinking

Inductive reasoning

- From specific examples to general principles
- Examples:
 - $8/1=8$; $27/1=27$; $118/1=118$;...
 - Division by 1 equals the number
 - $3+5=8$; $7+9=16$; $27+23=50$;...
 - Odd plus odd is even

Deductive reasoning

- From general principles to specific examples
- Example:
 1. When it rains things outside get wet
 2. The grass is outside
 3. When it rains, the grass gets wet

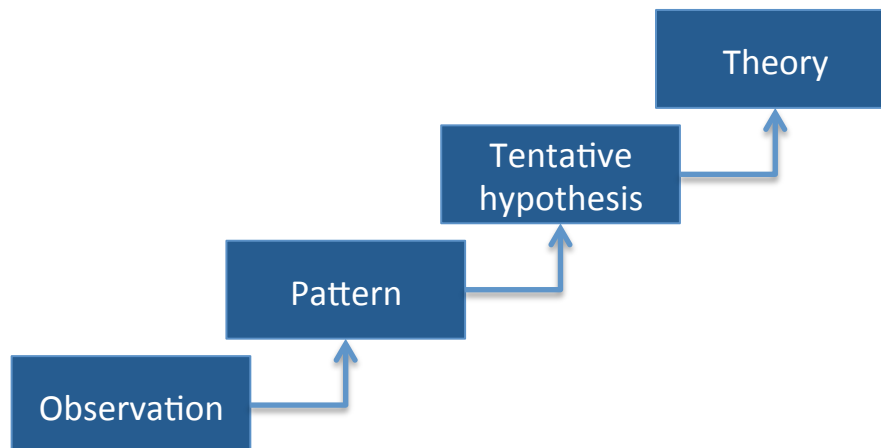
Concepts: Logical Thinking

Inductive reasoning

- From specific examples to general principles
- Examples:
 - $8/1=8$; $27/1=27$; $118/1=118$;...
 - Division by 1 equals the number
 - $3+5=8$; $7+9=16$; $27+23=50$;...
 - Odd plus odd is even

Deductive reasoning

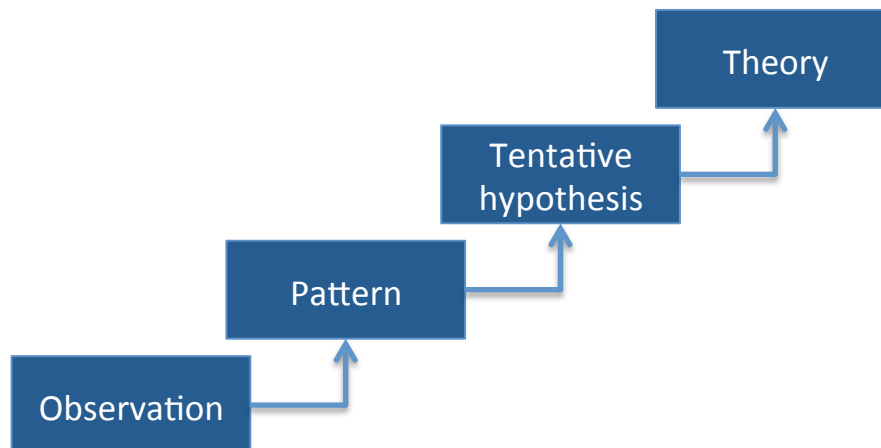
- From general principles to specific examples
- Example:
 - When it rains things outside get wet
 - The grass is outside
 - When it rains, the grass gets wet



Concepts: Logical Thinking

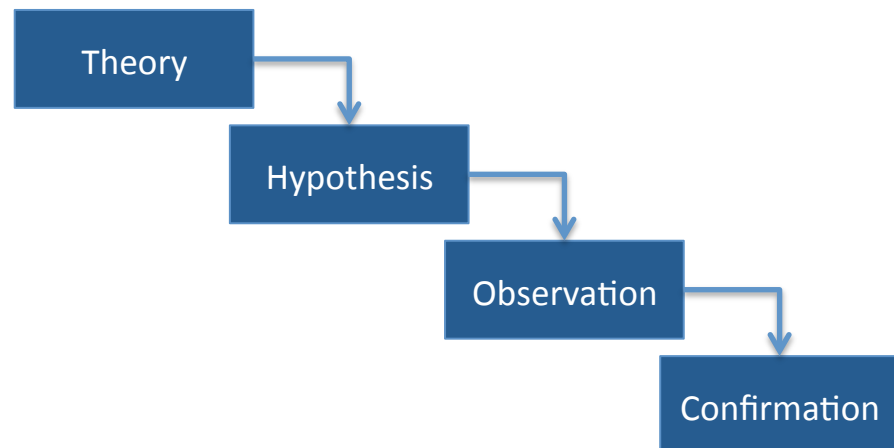
Inductive reasoning

- From specific examples to general principles
- Examples:
 - $8/1=8$; $27/1=27$; $118/1=118$;...
 - Division by 1 equals the number
 - $3+5=8$; $7+9=16$; $27+23=50$;...
 - Odd plus even is even



Deductive reasoning

- From general principles to specific examples
- Example:
 - When it rains things outside get wet
 - The grass is outside
 - When it rains, the grass gets wet



Concepts: Algorithms

- What is it?
 - A sequence of steps for solving a problem
- Why is it important?
 - Use algorithms in daily life to describe solutions to problems
 - Computers programs execute algorithms to perform specific tasks



Concepts: Algorithms

- What is it?
 - A sequence of steps for solving a problem
- Why is it important?
 - Use algorithms in daily life to describe solutions to problems
 - Computers programs execute algorithms to perform specific tasks



Course Goals

- Solve problems using decomposition and abstraction
- Use a programming language to express these solutions
 - Leave with a basic understanding of Python
- Be able to test those solutions using logical and deductive reasoning

The Course!

- Programming is learned by doing!
- Class will be a mix of lecture and lab:
 - Discuss concept
 - Apply concept
 - Repeat
- Labs that go unfinished must be turned in before the following class

Homework

- Periodic homework assignments
- Generally three class periods to complete
- Late assignments
 - -15% for day 1
 - -30% on day 2
 - No credit thereafter

Grading

Type	Overview	Weight
Labs	Every class	5
Homework	Every ~few weeks	20
Project	Group-based; final few weeks	25
Exams	Three: around weeks 5, 9, and 14	50

- Labs, homework, project: doing (coding)
- Exams: understanding (pencil/paper)

Integrity

- Don't cheat
 - Get help from friends and the Internet
 - Do not get answers from friends or the Internet
-
- Your education is up to you
 - This can be a very rewarding course if you try

Accessibility

- I'm here!
 - Catch me during office hours
 - Or just make an appointment
- Nabil is here too!

Email	jerome.white@nyu.edu	nabil.rahiman@nyu.edu
Office	A2-195 (or A2 1 st floor in general)	A2-182A (near the Design Technology Lab)
Hours*	Tue: 2:00 — 3:30 Tue: 3:00 – 4:00 Thu: 2:00 — 3:30 Sun: 3:00 – 4:00	Sun: 1:45 – 2:45

* Any changes will be announced

Basics of Computing

- A computer consists of
 - Input
 - Output
 - Storage
 - CPU
- Your phone is a computer



Programming

- “Instantiation” of computational thinking
- Sequence of instructions that specify the computation flow
 - Interpreted (very quickly) by the computer
 - Based on a formalization of a problem
- Instructions are specified in a given “language”

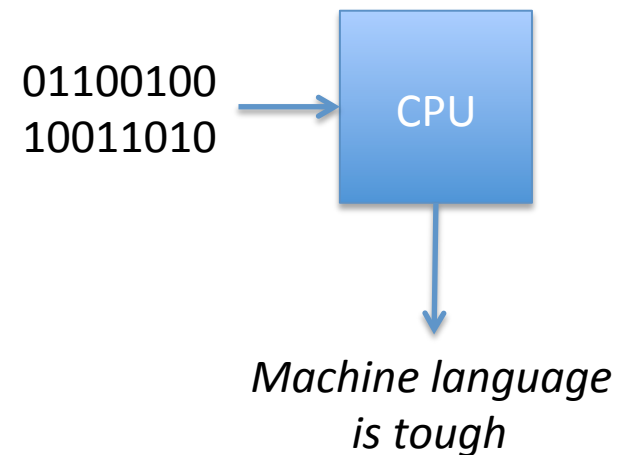
Programming Languages

- Programming versus natural languages

	Natural languages	Programming languages
Ambiguity	Rampant	Minimal
Redundancy	Lots	Almost none
Literalness	Room for interpretation	Means what it says

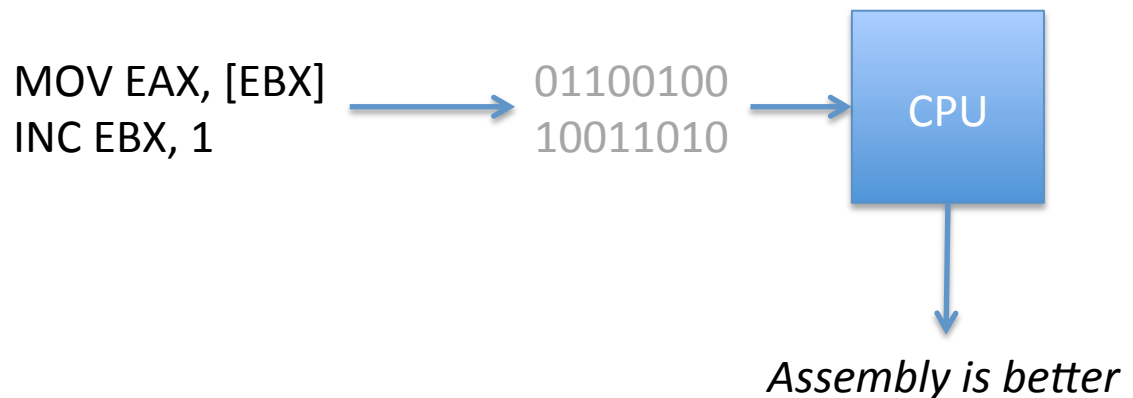
Programming Languages

- Programming versus natural languages
- High-level versus low-level languages



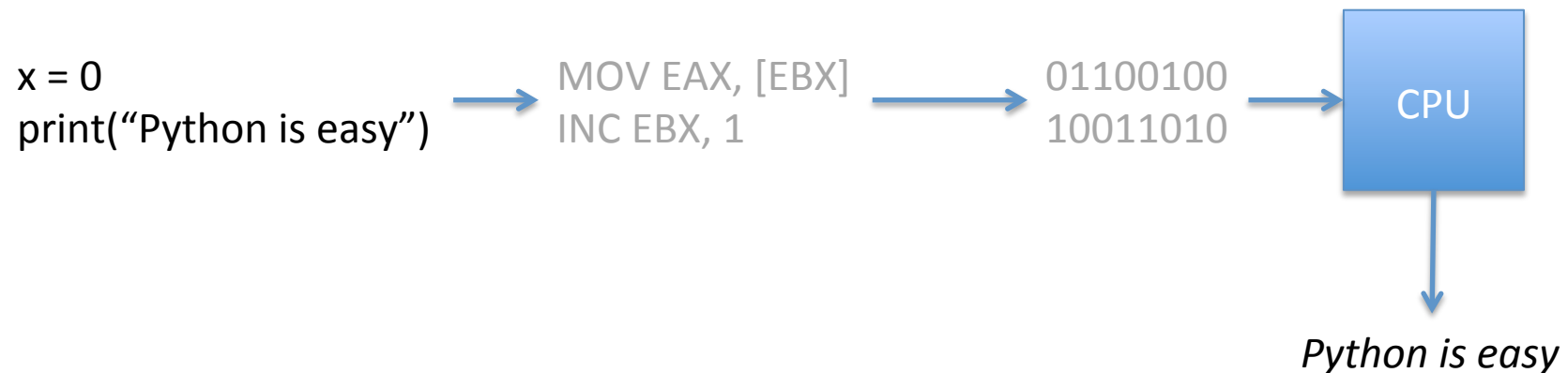
Programming Languages

- Programming versus natural languages
- High-level versus low-level languages



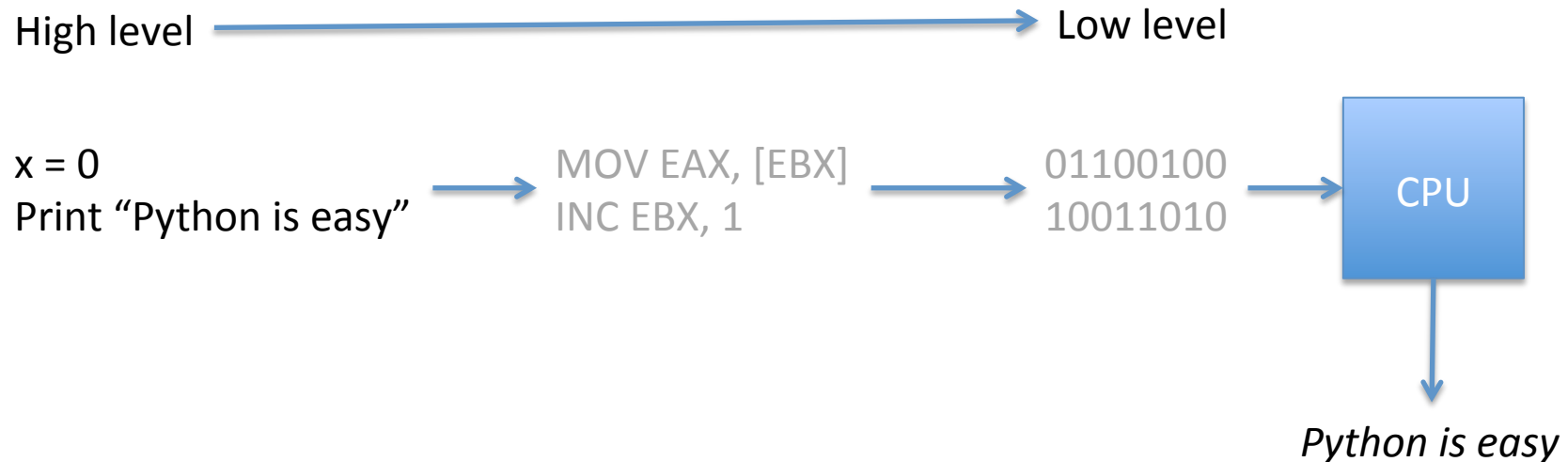
Programming Languages

- Programming versus natural languages
- High-level versus low-level languages



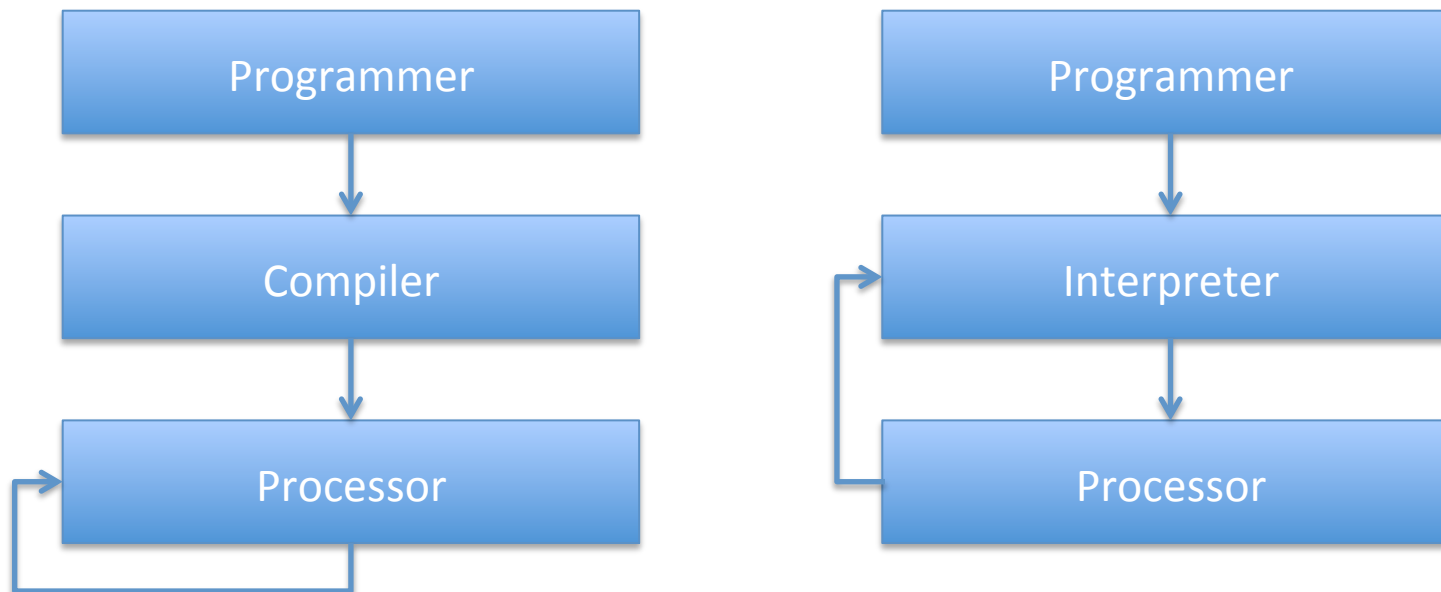
Programming Languages

- Programming versus natural languages
- High-level versus low-level languages



Programming Languages

- Programming versus natural languages
- High-level versus low-level languages
- Compiled versus interpreted languages



Python

- High-level, interpreted language
 - Means do exist to compile/executable
- Designed for rapid prototyping and easily maintainable code
- Comes in two “flavors”: 2-series and 3-series
 - 2-series more mature, 3-series more “modern”
 - We’ll be using the 3-series

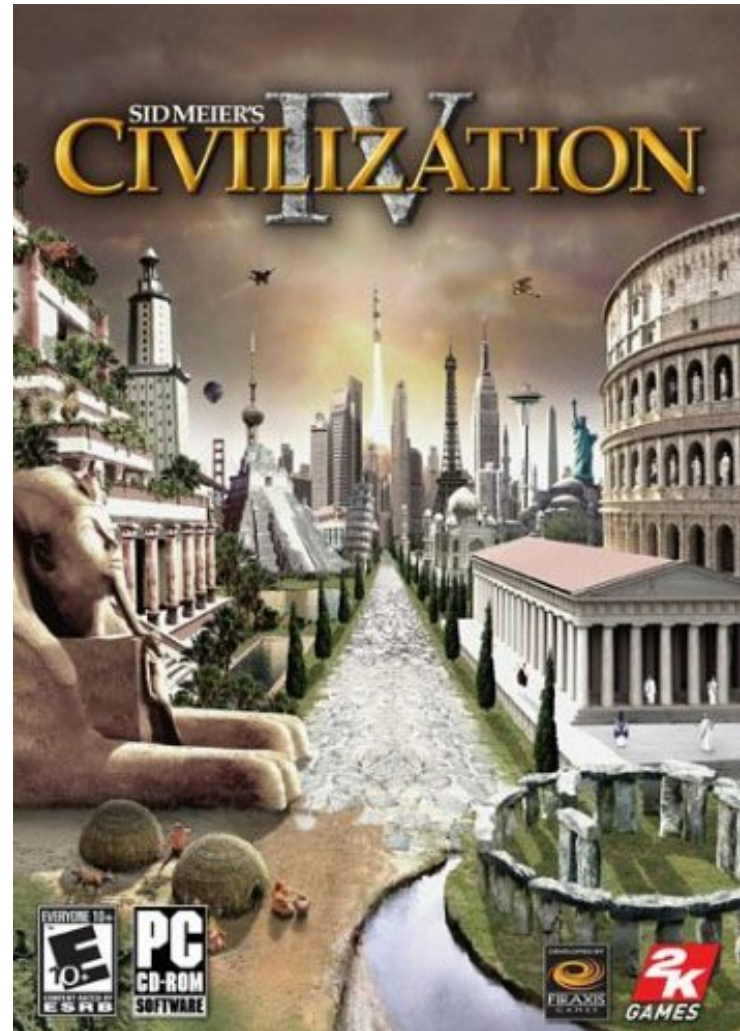
Python is Useful

- Actively developed and supported
 - Large number of extensions that make your life easier
- Prominent language at Google
 - Also Netflix and NASA
- Used to develop Civ4



ضربة حظ

Google بحث



Your turn!

1. Install Anaconda/Spyder
<https://www.continuum.io/downloads>
2. Print your name
3. Use one line of the interface to calculate:
 - If you run a 10 kilometer race in 43 minutes 30 seconds what is your average time per mile? (Assume 1.609 kilometers to a mile.)^{[1](#)}
 - What is your average speed in miles per hour?^{[2](#)}
 - The campus network supports downloads of 1.5 megabytes per second. How long will it take to download a 500 gigabyte file? (Assume 1000 megabytes per gigabyte.)^{[3](#)}
4. Repeat 2 and 3 by creating and executing a Python file

Programming Syntax

- Rules defining how elements of an instruction should be combined

	English	Python
Incorrect	“John chess plays” (noun-noun-verb)	3 3 + (operand-operand-operator)
Correct	“John plays chess” (noun-verb-noun)	3 + 3 (operand-operator-operand)
	<ul style="list-style-type: none">• Syntax errors, to an extent, are tolerable in natural languages• Our brains are smart	<ul style="list-style-type: none">• Syntax errors are a show-stopper in computing• Computers aren’t that smart

Programming Semantics

- The *meaning* of syntactically legal structures
- Programming language design tries to remove ambiguities, but the onus is on the programmer

English	Python
"Colorless green ideas sleep furiously"	2+3+5/3
<ul style="list-style-type: none">• Doesn't make sense• Humans have the ability to reason about, and accept, such errors	<ul style="list-style-type: none">• Python designers have made an underlying assumption about operator order• Program can still run, but the output is unexpected

Errors

- Errors in syntax and semantics constitute most of the problems you will run into

Type	Meaning	Example
Syntactic	<ul style="list-style-type: none">• Used characters, or a sequence of characters, that are illegal in the language	<code>print('hello'}</code>
Semantic	<ul style="list-style-type: none">• Unexpected results based on a misunderstanding of the meaning of the operation	$3 + 4 + 1 / 3$
Run-time	<ul style="list-style-type: none">• Errors that occur during the course of a running program• Undeclared variables usually the biggest culprit• Lots of current research around predicting these errors before-hand	$x = 0$ $1 / x$

Your turn!

- Python 3 has two ways of dividing numbers:
 1. `x / y`
 2. `x // y`
 - What's the difference?
 - What's going on?
- Replace `/` with `%`
 - What's happening here?
- “Importing” allows you to others code in your programs
 - Try `import this`
 - Try `import antigravity`

Moving Forward

○ So a fancy calculator is what runs Google?

➤ Yes, sort of...

<i>Layered Simplicity</i>		
Building blocks	• Data types (starting next class)	How data is organized and stored
	• Algorithms (starts next month)	The instructions to get things done
Execution model	<ol style="list-style-type: none">1. Sequence,2. Selection, and3. Repetition	

Intro to Computer Science

Previous

- What is computer science
- Programming languages
- Syntax v. Semantics
- Python!

Next

- Types
 - Introspection
 - Casting
- Variables
- Operators
- Input
- String methods

Readings

Gaddis

- Chapter 1
- Appendix A
- Appendix B

Readings

Gaddis

- Chapter 2
- Chapter 8.3*

Python Std. Lib.

- [Section 4.7.1](#)

* “Searching, Manipulating”