Outline Sept 5:

• Introductions
• Areas of computer science
• Nonograms followup

• Intro to **python3** and the interpreter
• Variables, values, assignment
• Types and conversion: `int`, `float`, `str`
• `input`, `print` built-in functions

Reminder: visit 249 (my office) as part of Lab 0!
Areas of Computer Science
Computer Science Research Areas

- Artificial Intelligence
- Computer Architecture
- Computational Biology
- Databases
- Computer Science Education
- Computer Graphics

- Human-Computer Interaction
- Operating Systems
- Programming Languages
- Scientific Computing
- Cyber Security
- Theory

Credit: UC Berkeley EECS, https://www2.eecs.berkeley.edu/Research/Areas/CS/
Artificial Intelligence

Computer Architecture

Computational Biology

Figure credit: “Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls” by the Wellcome Trust Case Control Consortium
Databases

Figure credit: “Distributed transactions across cloud databases”
– Microsoft Azure Database

Computer Graphics

“Adaptive tissue modeling”
Vidal et al, 2006

“Zootopia”
Disney, 2016

Computer Science Education
Human-Computer Interaction

Universal ATM Interface by Kristin Suzanne Bessette

Programming Languages

Operating Systems

Universal ATM Interface by Kristin Suzanne Bessette

Wikimedia Commons (by Golftheman)
Scientific Computing

“A numerical solution to the heat equation on a pump casing model using the finite element method.” – Wikipedia

Cyber Security

Credit: Hugh Boyes (2014)

Theory

“An artistic representation of a Turing machine. Turing machines are used to model general computing devices.” – Wikipedia
Nonogram followup
Handout example

Image credit: www.nonograms.org
Systematic solving of a nonogram
Systematic solving of a nonogram
Systematic solving of a nonogram
Solution
Algorithm for checking row solution
Algorithm for checking row solution

true solution

2 3 1

(E E F F E E E F F F E E F F)

x = 0 (variable to keep track of F blocks)

user solution
Algorithm for checking row solution

true solution

2 3 1

x = 0 (variable to keep track of F blocks)

user solution
Algorithm for checking row solution

true
solution

2  3  1

user
solution

x = 1  (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

2 3 1

user solution

E E F F E E E F F F E E F F

x = 2 (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

user solution

\[ x = 0 \]  (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

2 3 1

user solution

2

x = 0 (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

2 3 1

user solution

2

x = 0 (variable to keep track of F blocks)
Algorithm for checking row solution

x = 1 (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

2 3 1

user solution

2

x = 2 (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

user solution

2 3 1

2

x = 3 (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

user solution

2 3 1

x = 0 (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

2 3 1

user solution

2 3

x = 0 (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

\[
\begin{array}{ccc}
2 & 3 & 1 \\
\end{array}
\]

user solution

\[
\begin{array}{ccc}
2 & 3 \\
\end{array}
\]

\[x = 1\] (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

2 3 1

user solution

2 3

x = 2

(variable to keep track of F blocks)
Algorithm for checking row solution

$x = 0$ (variable to keep track of F blocks)
Algorithm for checking row solution

true solution

user solution

2 3 1

≠

2 3 2

x = 0 (variable to keep track of F blocks)

OUTPUT: NO
Python interpreter: demo
Key CS concepts today

- **Variables** as a way to store **values**

- **Assignment operator** (\(=\)) is a way to change the value of a variable (not symmetric like equals operator in math!)
  - Variable name on the left, expression on the right

- The **type** of a variable is the type of the value it refers to

- We can **convert** a variable to a different type, but it does not change the value of the original variable
Functions for today

- type()
- input()
- int()
- str()
- float()

Types for the first few weeks

- int
- float
- str
Types
- int
- float
- str