

CONCEPTS OF PROGRAMMING

©TTAA .BB INKKOWSKJ ,22019

MPCS 50101
MODULE 1



THE UNIVERSITY OF
CHICAGO

WELCOME TO COMPUTING

© T.A. BINKOWSKI, 2020

MODULE 1
MPCS 50101



THE UNIVERSITY OF
CHICAGO

COURSE LOGISTICS

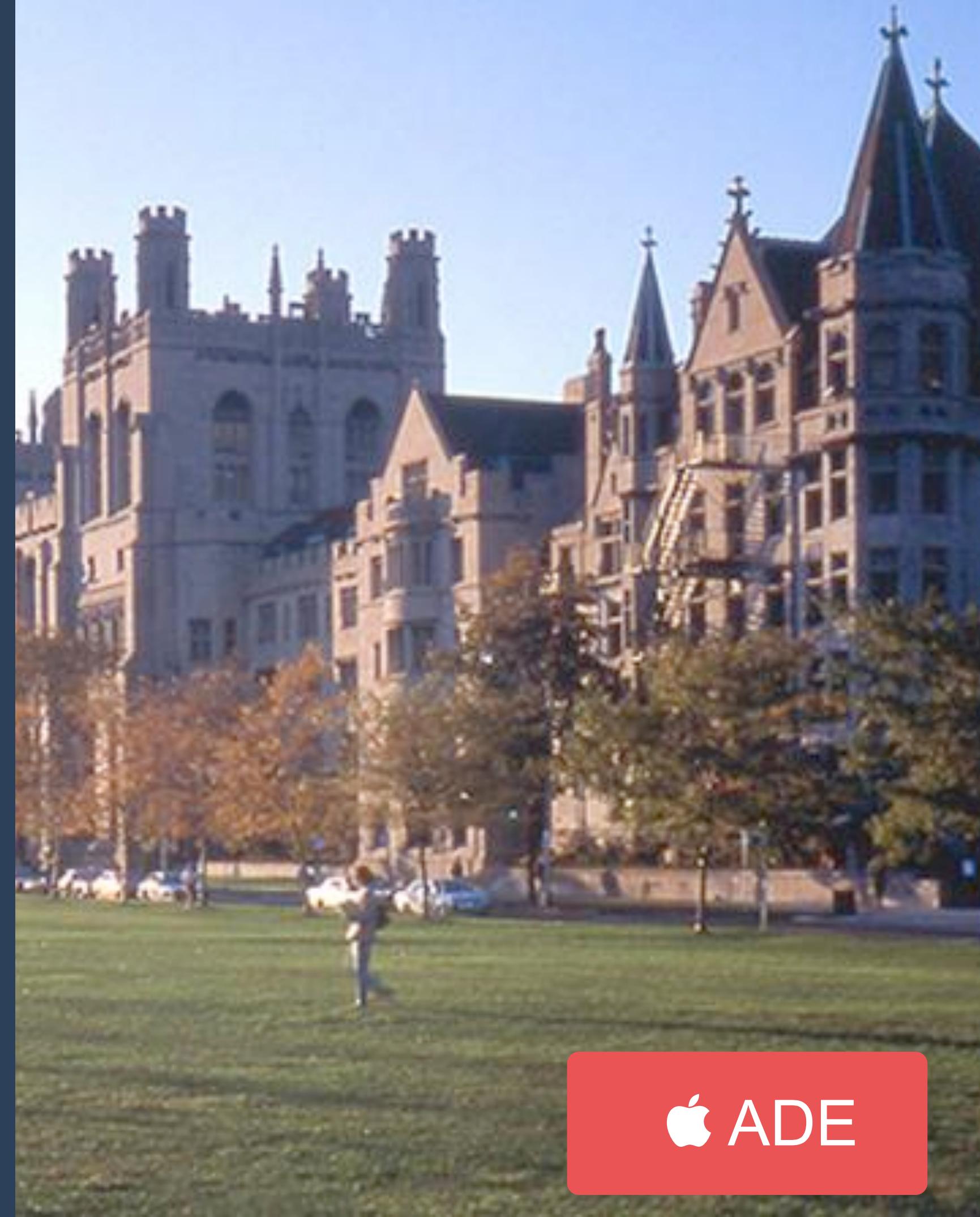
- MPCS 50101 - Concepts of Programming
 - This should be your first programming class
- Prerequisites
 - Desire
 - Patience
 - Persistence



COURSE LOGISTICS

COURSE LOGISTICS

- Instructor
 - Andrew Binkowski
 - The University of Chicago
 - Center for Structural Genomics of Infectious Diseases, Computation Institute
 - MPCS: Mobile Computing, Bioinformatics
 - ANL: Midwest Center for Structural Genomics, Leadership Computing Facility



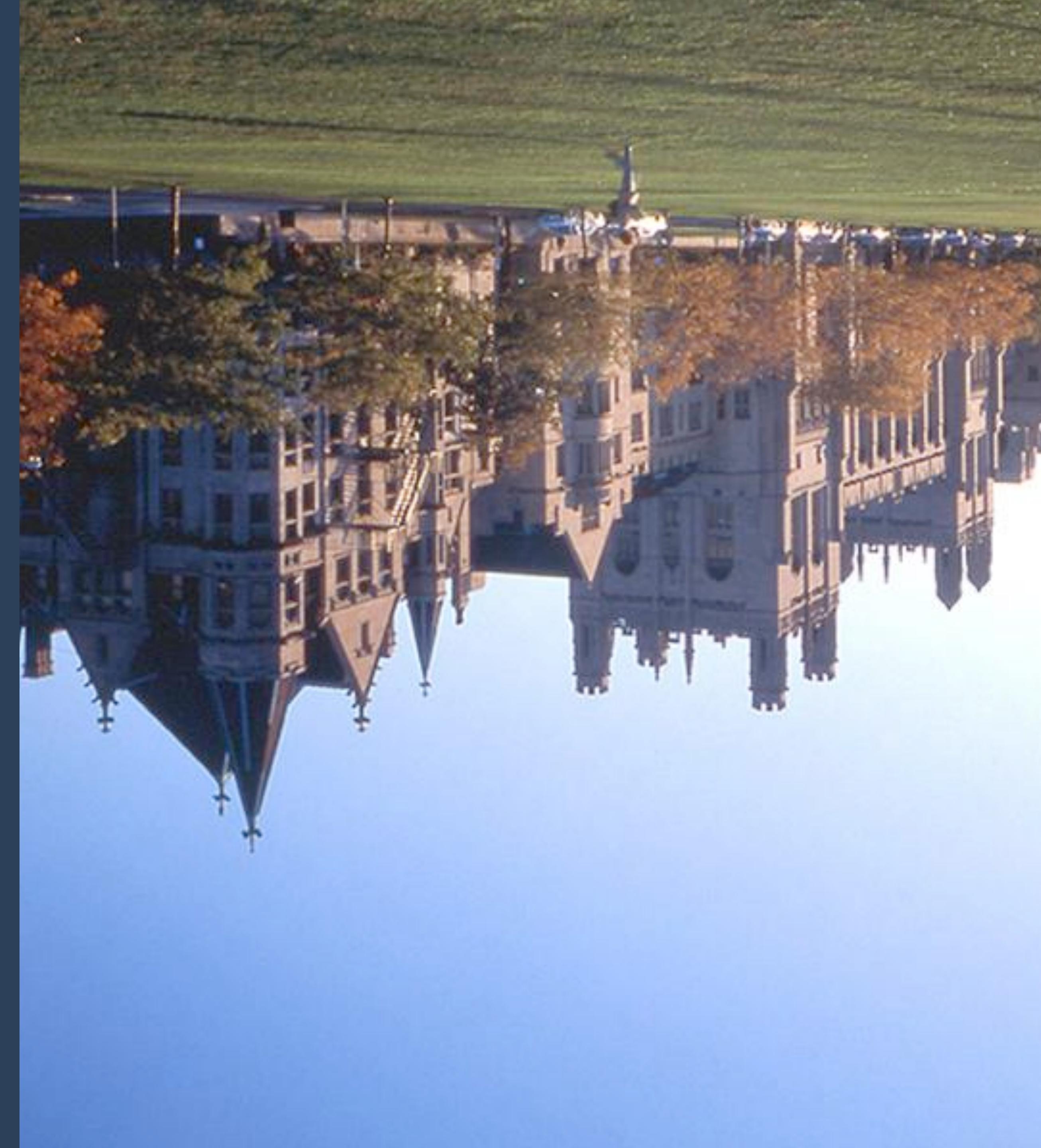
COURSE LOGISTICS

YOUR BEST RESOURCES
FOR THIS COURSE

- Teaching Assistants
 - Jiamao Zhang, MPCS Alumni, Senior Data Engineer at Walgreens

COURSE LOGISTICS

- Flipped Classroom
 - Videos of lectures
 - Meet for discussions, demos and office hours



COURSE LOGISTICS



Account

Dashboard

Courses

Calendar

Inbox

Commons

Help

MPCS 50101 1 Concepts of Programming > Modules

2020.04

Collapse All

View Progress

+ Module

⋮

Home

Announcements



Syllabus

Modules



Assignments

Discussions

Library Reserves

People

Grades

Panopto Video

Purchase UChicago
Bookstore Course
Materials

Module 1: Welcome to Computing

Module 1: Overview

Resources

Module 1: Breakout Exercises

Module 1: Lecture Slides

Lecture Videos

Module 1: Welcome to Programming

ORGANIZED BY
MODULES

RELEASE WEEKLY

COURSE LOGISTICS

- Synchronous Lecture
 - Monday 5:30 ~ 7:00 PM (CST)
 - Lectures will recorded and available throughout the quarter
- Attendance is highly recommended

GREETINGS AND OUTLINE (5 MIN)

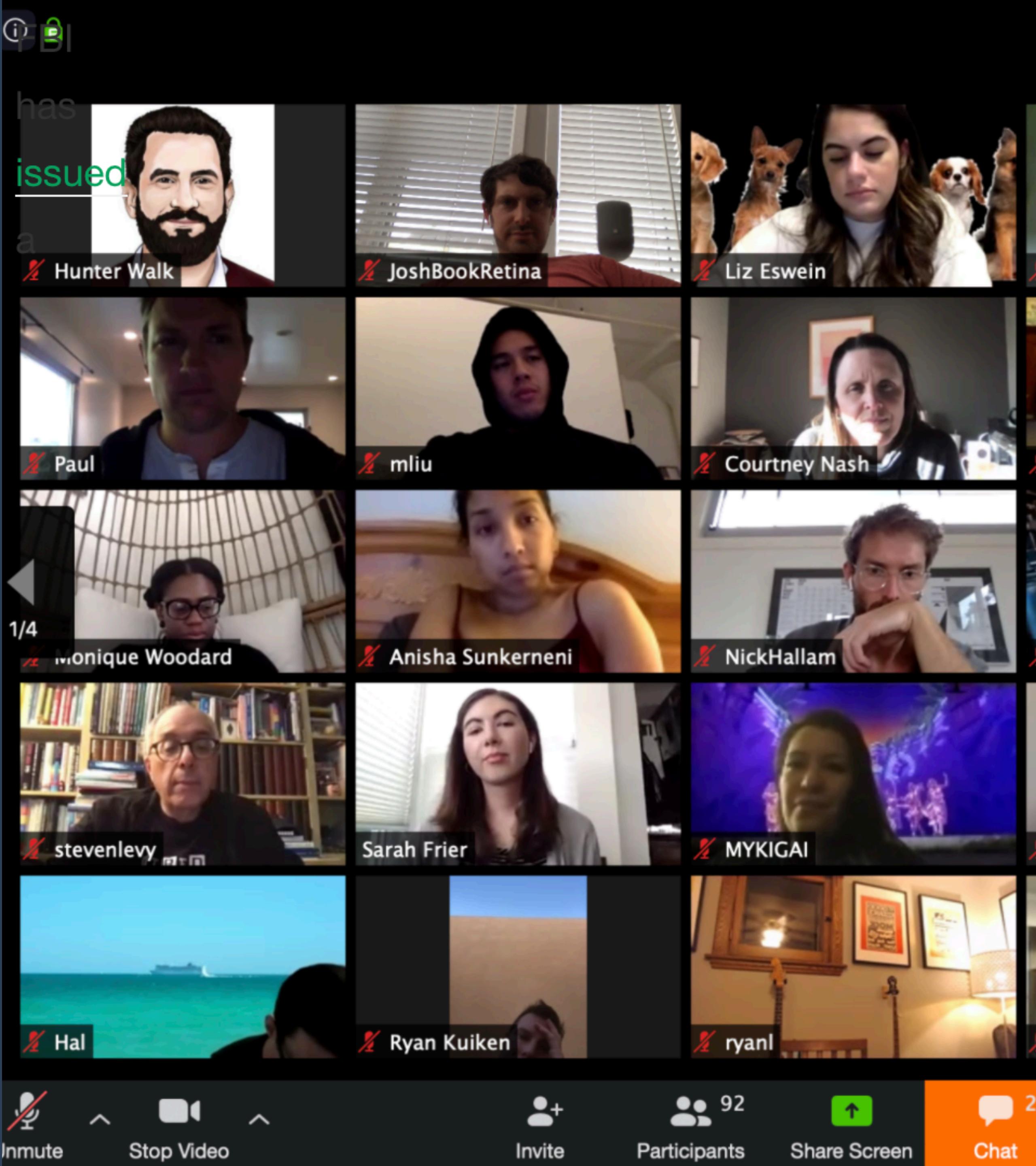
REVIEW "LECTURE QUIZ" (10 MIN)

BREAKOUT ROOM EXERCISES (30 MIN)

MODULE PREVIEW (45 MIN)

COURSE RESOURCES

- Zoom Etiquette
 - Mute if you are not talking
- Questions
 - "Raise Hand"
 - Type it in the Zoom chat
 - Post in Slack module
- All sessions will be recorded
 - You can opt-out by turning camera and microphone off
 - Will not be distributed outside of class



COURSE RESOURCES

- I would appreciate if you turn on camera or set profile picture of yourself

The image shows a video conference interface. At the top, there is a small thumbnail of a man with a beard, identified as "Hunter Walk". Above his thumbnail, the text "has issued a" is displayed, with "issued" in green. Below his thumbnail, there is a red icon and the name "Hunter Walk". At the bottom, there is a larger thumbnail of a woman with long dark hair, identified as "MYKIGAI". Behind her thumbnail is a purple background with some faint, blurry figures. Below the thumbnails, there is a control bar with icons for microphone (muted), video (stop video), and participant count (92). There are also buttons for "Invite", "Participants", "Share Screen", and "Chat".

COURSE LOGISTICS



- Account
- Dashboard
- Courses
- Calendar
- Inbox
- Commons
- Help

MPCS 50101 1 Concepts of Programming > MPCS 50101 1 (Autumn 2020) Concepts of Programming

2020.04

Search in folder "Module 1"...



Create ▾

Module 1 ▾

Sort by: Name Duration Date ▾

+ Add folder



Module 1: Setting Up A Development Environment

11 hours ago



Module 1: Welcome to Programming

12 hours ago

ASYNCHRONOUS
LECTURES
USING
PANOPTO
VIDEO

COURSE LOGISTICS

- Slack for **ALL** class communication
 - MPCS50101-2021-summer
- Post questions, interesting links, etc.
 - Anyone can post
 - Anyone can answer
- Personal communication via email

The screenshot shows a Slack interface with the following details:

- Channel Header:** mpcs50101-20... (dropdown), Andrew (user icon)
- Header Buttons:** Jump to..., <, >
- Left Sidebar:** Threads, Apps, Channels (with a plus sign), # general (highlighted in green), # git, # office-hours, # random.
- Direct Messages:** Slackbot, Andrew (you), Anirudh Sriram (partially visible).
- Message Log:**
 - September 30th, 2021:** Andrew joined #general along with Yixiao.
 - October 1st, 2021:** Andrew: Welcome to Concepts of Comput...
 - Reactions: thumbs up (2), smiley face.
 - Bottom:** Session 1 lectures slides. PDF ▾

COURSE LOGISTICS

⌘2

EC

⌘3

M

⌘4



⌘5

m

⌘6

+

Channels



assignment1

general

random

Direct Messages



slackbot

Andrew (you)

Jane Makin

+ Invite People

Apps



#general

You created this channel today. This is the very beginning of the workspace-wide communication and announcements. All members

[+ Add an app](#) [👤 Invite people to mpcs50101-2018-winter](#)

TRY TO USE THREADS
TO REPLY TO SPECIFIC
POSTS

Today



Andrew 6:36 PM
joined #general.

Start a thread



Andrew 6:39 PM ☆
Welcome to Concepts of Programming!



Jane Makin 6:40 PM
joined #general.

COURSE LOGISTICS

- Turn in work via Canvas assignments
- Github for coursework and exams
 - Commit your repo to “turn in”
 - Private repositories

The screenshot shows the Classroom for GitHub interface. At the top, there's a header with the GitHub logo and a 'classroom.github.com' URL. Below the header, a green banner reads 'Your course assignments on GitHub'. It explains that Classroom for GitHub automates repository creation and access control, making it easy to distribute starter code and collect assignments on GitHub. A blue button invites users to 'Sign in with your GitHub account to get started'. To the right, there are sections for 'Invite students' (using unique invitation URLs) and 'Review progress' (teachers can browse submissions by assignment). A red callout box labeled 'LATER IN THE COURSE' points towards these features.

LATER IN THE COURSE

Your course assignments on GitHub

Classroom for GitHub automates repository creation and access control, making it easy to distribute starter code and collect assignments on GitHub.

Sign in with your GitHub account to get started

Invite students

Students gain access to assignments using unique invitation URLs, eliminating the need to collect GitHub usernames in advance and manage access manually.

Review progress

Teachers and teaching assistants can browse submissions by assignment, making it easy to follow along with student progress.

with ❤ by GitHub

Classroom for GitHub is open source.

COURSE LOGISTICS

- Andrew
 - Wednesday 10:00AM (CST)
- Jiamao
 - Friday 6:00 - 9:00PM (CST)
- Thomas
 - Weeknight TBD
 - Weekend TBD

OFFICE HOURS ARE
YOUR BEST
OPPORTUNITY TO
LEARN

#OFFICE-HOURS
UPDATES, CHANGES,
QUESTIONS

COURSE DESIGN

COURSE DESIGN

- Provide the foundation for you to be successful in the program
 - Think like a computer scientist
 - Introduce best practices in programming
 - Expose you to tools and technologies that are part of a computer scientist's toolkit

COURSE DESIGN

START THINKING AND
TALKING LIKE A
COMPUTER SCIENTIST



COURSE DESIGN

- This will be funny in just a few weeks

MAN, I SUCK AT THIS GAME.
CAN YOU GIVE ME
A FEW POINTERS?

|

0x3A28213A
0x6339392C,
0x7363682E.

I HATE YOU.

/



COURSE DESIGN

- Challenges
 - Different backgrounds of students
 - I don't know what you don't know
 - Pacing of the course
- Please ask questions frequently



SYLLABUS

SYLLABUS

CONCEPTS OF PROGRAMMING

MPCS 50101



CANVAS

Welcome to computing! In this course , we provide an immersive introduction to programming for students who are new to computer science. We use the Python programming language to explore general-purpose computer programming and algorithmic reasoning. Tools and technologies used throughout the program will be introduced. The aim of this course is to provide a solid foundation that will ensure student success in the Masters in Computer Science program.

Instructors

- Andrew Binkowski, PhD., Asst. Clinical Prof. University of Chicago
 - abinkowski@uchicago.edu

SYLLABUS

CANVAS

- Topics to be covered
 - Modules 1-5
 - Construct algorithms, and express them as computer programs
 - Refine and improve programs by an iterative process
 - Modules 6-9:
 - Core principles of object-oriented programming
 - Fundamental data structures
 - Basics of computational efficiency and "Big-O" notation

SYLLABUS

Module 1: Overview

Outline

- Welcome to Computing
 - Course Logistics, Design, Technologies and Resources
- Modern Computing and Computational Thinking
 - Everything You Need to Know About Programming
 - Components of Every Computer Program
- Welcome to Programming
 - Programming Languages
 - Programming Paradigms
 - Debugging
- Setting Up a Development Environment
 - Shell
 - Text Editor
 - Programming Language
 - Your First Program in Python
 - Package Manager
 - Debugger
 - Version Control

Reading List

- [Automate the Boring Stuff](#) ↗ Chapters 0 and 1.
- [Think Python](#) ↗ Chapters 1 and 2
- [Command Line Crash Course \(LPTHW\)](#) ↗

Links

- Books
 - [Think Python](#) ↗
 - [Learn Python the Hard Way](#) ↗ .
 - [Official Python Documentation](#) ↗
 - [WikiBooks Python](#) ↗
- Development environment

SYLLABUS

Account	27	28	29	30	1	2	3	11 18 25	12 19 26	13 20 27	14 21 28
Dashboard											
Courses											
Calendar	4	5	6	7	8	9	10				
Inbox											
History											
Commons	11	12	13		15	16	17				
Help											
	18	19	20		22	23	24				
	25	26	27		29	30	31				

CANVAS
CALENDAR
WILL HAVE
IMPORTANT
DATES

5:30p Midterm Exam

▼ CALENDARS

- T. Andrew Binkows
- MPCS 50101 1 (Su)
- Concepts of Programmi
- MPCS 50101 1 (Au)
- Concepts of Programmi
- MPCS 50101 1 (Wi)
- Concepts of Programmi

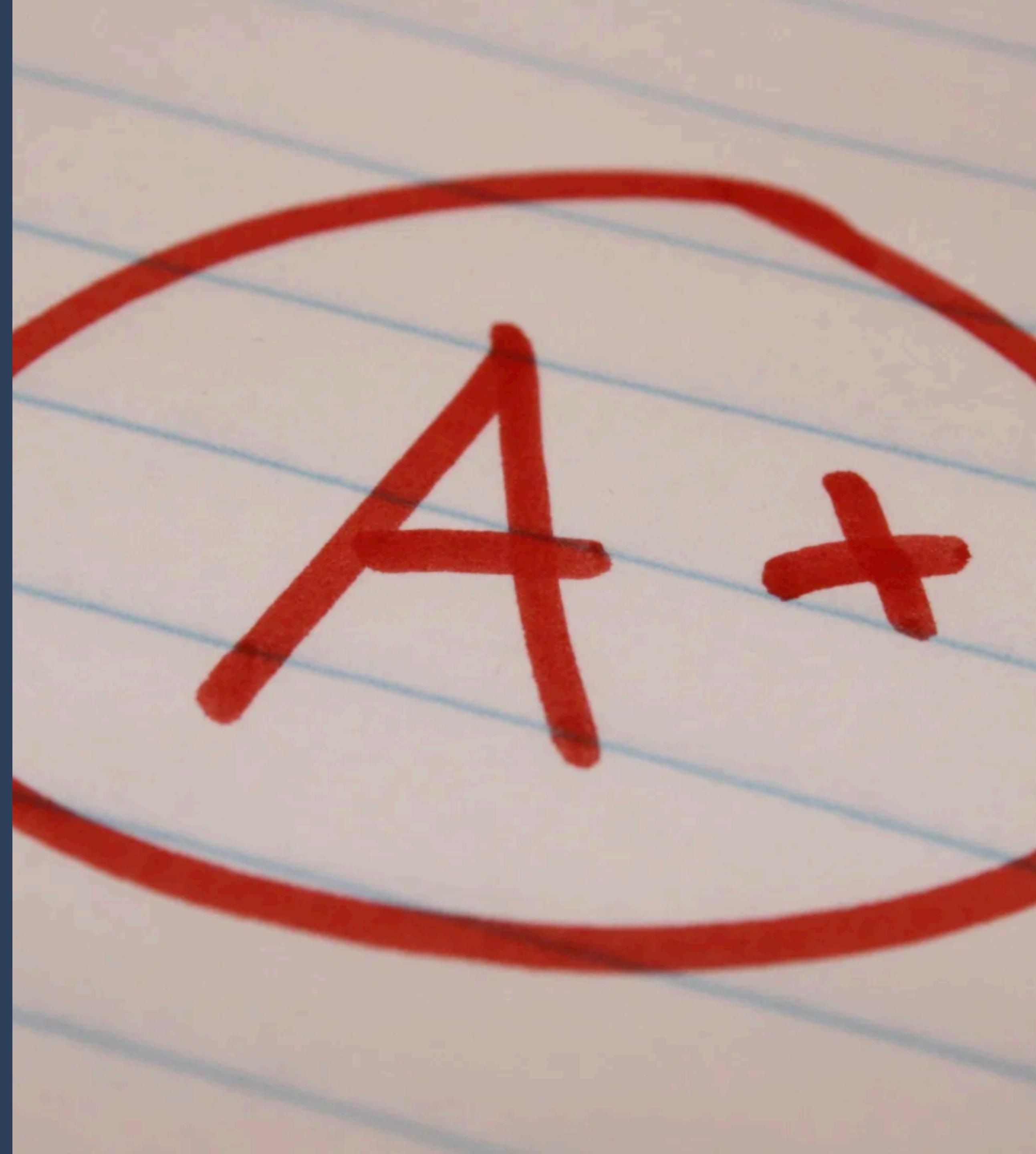
► UNDATED

- Calendar Feed

COURSE WORK

COURSE WORK

- Final grade will be determined as follows:
 - 75% assignments
 - 5% midterm exam
 - 10% final project
 - 5% lecture quiz
 - 5% participation



COURSE WORK

- Assignments
 - The bulk of the coursework (practice makes perfect)
 - Each week builds on the previous, important to keep up
 - Work is (will be) submitted through GitHub
 - No late work is accepted (unless extraordinary circumstances, less than 24 hours)
 - You may resubmit assignments one time and split the difference

ITERATION AND
IMPROVEMENT

SYLLABUS

- Midterm Exam
 - Check point to make sure that everyone is on the same page
- Final Project
 - Overview of entire course material

COURSE WORK

- Lecture Quiz
 - Weekly questions that come from the lecture videos and reading
 - Due before next class
 - Highlight important topics that are not covered in assignment
 - Retake as many times as you want

Quiz Instructions



Question 1

CANVAS

Which of the following criteria that is used to evaluate "good" program considered objective?

- Efficiency
- Readability
-
- Style



Question 2

1 pts

Which of the following are true about Python:

- It is an interpreted language.
- It can be run in script mode.
- It can be run in interactive mode.
- It is a compiled language.



Question 3

1 pts

Since the release of Python 3, Python 2 is obsolete.

COURSE WORK

- Participation
 - To get the most out of remote learning try to engage with the instructors and classmates as much as possible
 - Full credit
 - Being active in Slack
 - Working collaboratively in breakout rooms
 - Making appearance in office hours



COURSE WORK

- Honor Code
 - All the assignments should be your own work
 - Department policies are strictly enforced
 - Cite any resources you use on homework
 - Includes online resources (Stack Overflow, blogs, GitHub, etc.)



COURSE WORK

- Programming can be frustrating
 - Don't be afraid to take a break
 - Go for a walk...it really helps

6 hours later



COURSE WORK

- Every mistake is a learning opportunity (to learn not to do it again)
- The more mistakes you make the more you will learn 🤔

6 hours later



COURSE TECHNOLOGIES

COURSE TECHNOLOGIES

- Developing computer program requires a variety of tools and technologies
- We will use “popular” technologies that you will see throughout the program and industry

```
proteinworks -- abinkows@miralac1:~ more 127x77
ows@miralac1:~ bash bash

number(txt):
    ]
    txt.splitlines():
    startswith("ATOM"):
        number = int(line[7:11])
        [atom_number] = line

        orted(lines):
            .append(lines[line])
            oin(new_lines)

        db_txt, tag_func):
    ]
    db_txt.splitlines():
    nc(line):
    nue
    .append(line)
    oin(new_lines)

ension(filename):
h.splitext(os.path.basename(filename))[0]

ce(atom1_xyz,atom2_xyz):
    is a list [x,y,z] coordinate """
    sqrt((atom1_xyz[0]-atom2_xyz[0])**2+
        (atom1_xyz[1]-atom2_xyz[1])**2+
        (atom1_xyz[2]-atom2_xyz[2])**2)

ine(line):
e[30:38])
e[38:46])
e[46:54])
]

neighbors(pdb_txt,ligand_xyz,cutoff):
]
db_txt.splitlines():
startswith("ATOM"):
    in = xyz_from_pdpline(line)
    ligand_atom in ligand_xyz:
    dist = atomic_distance(protein,ligand_atom)
    if dist < cutoff:
        #print "%s - %f" % (line,dist)
        new_lines.append(line)
    lines

d_coordinates(pdb_txt,ligand_key):
    [x,y,z] list of the coords of a given ligand
    ionally print to file?

tracting coordinates for ligand key "
key

db_txt.splitlines():
startswith("HETATM"):
    type = (line[17:20]).strip()
    _id = line[21]
    num = int(line[22:26])
    t ligand_key
```

COURSE TECHNOLOGIES

- Command line tools using Bash, Powershell
- Programming with Python
- Text editing with Visual Studio Code
- Version control using Github
- Package management with Anaconda

```
proteinworks — abinkows@miraclac1:~ — more — 127x77
abinkows@miraclac1:~ bash

import os
import sys
import math

def sort_by_atom_number(txt):
    lines = {}
    new_lines = []
    for line in txt.splitlines():
        if line.startswith("ATOM"):
            atom_number = int(line[7:11])
            lines[atom_number] = line

    for line in sorted(lines):
        new_lines.append(lines[line])
    return '\n'.join(new_lines)

#-
def strip_lines(pdb_txt, tag_func):
    new_lines = []
    for line in pdb_txt.splitlines():
        if tag_func(line):
            continue
        new_lines.append(line)
    return '\n'.join(new_lines)

#-
def strip_pdb_extension(filename):
    return os.path.splitext(os.path.basename(filename))[0]

#-
def atomic_distance(atom1_xyz,atom2_xyz):
    """ atom1_xyz is a list [x,y,z] coordinate """
    return math.sqrt((atom1_xyz[0]-atom2_xyz[0])**2+
                    (atom1_xyz[1]-atom2_xyz[1])**2+
                    (atom1_xyz[2]-atom2_xyz[2])**2)

#-
def xyz_from_pdbleline(line):
    x = float(line[30:38])
    y = float(line[38:46])
    z = float(line[46:54])
    return [x,y,z]

#-
def extract_atom_neighbors(pdb_txt,ligand_xyz,cutoff):
    new_lines = []
    for line in pdb_txt.splitlines():
        if line.startswith("ATOM"):
            protein = xyz_from_pdbleline(line)
            for ligand_atom in ligand_xyz:
                dist = atomic_distance(protein,ligand_atom)
                if dist < cutoff:
                    #print "%s - %f" % (line,dist)
                    new_lines.append(line)
    return new_lines

#-
def extract_ligand_coordinates(pdb_txt,ligand_key):
    """ Return an [x,y,z] list of the coords of a given ligand
    TODO: Optionally print to file?
    """
    print "### Extracting coordinates for ligand key "
    print ligand_key

    coords = []
    for line in pdb_txt.splitlines():
        if line.startswith("HETATM"):
            res_type = (line[17:20]).strip()
            chain_id = line[21]
            res_num = int(line[22:26])
            #print ligand_key
            coords.append([res_type,chain_id,res_num])

    return coords
```

COURSE TECHNOLOGIES

- Other technologies
 - Jupyter Notebooks
 - linux.uchicago.edu
 - External APIs
 - Python Debugger
 - VS Code packages
 - Continuous Integration

The screenshot shows a Jupyter Notebook interface with the title "jupyter covid_19_dashboard Last Checkpoint: Last Friday at 11:45 PM (u)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with various icons for file operations and cell execution. The notebook contains several code cells:

- In [13]:

```
# importing libraries

from __future__ import print_function
from ipywidgets import interact, interactive, fixed,
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets
```
- In [14]:

```
# loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_deaths.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_confirmed.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_recovered.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_country_metadata.csv')
```
- In [15]:

```
confirmed_df.head()
```
- In [16]:

```
recovered_df.head()
```
- In [17]:

```
death_df.head()
```
- In [18]:

```
country_df.head()
```

COURSE RESOURCES

COURSE RESOURCES

- No required text books
 - Books are good for self learning
 - Do not match the scope of this course
- We will use open source materials throughout the course

Reading List

CANVAS

- [Automate the Boring Stuff](#) ↗ Chapters 0 and 1.
- [Think Python](#) ↗ Chapters 1 and 2
- [Command Line Crash Course \(LPTHW\)](#) ↗

Links

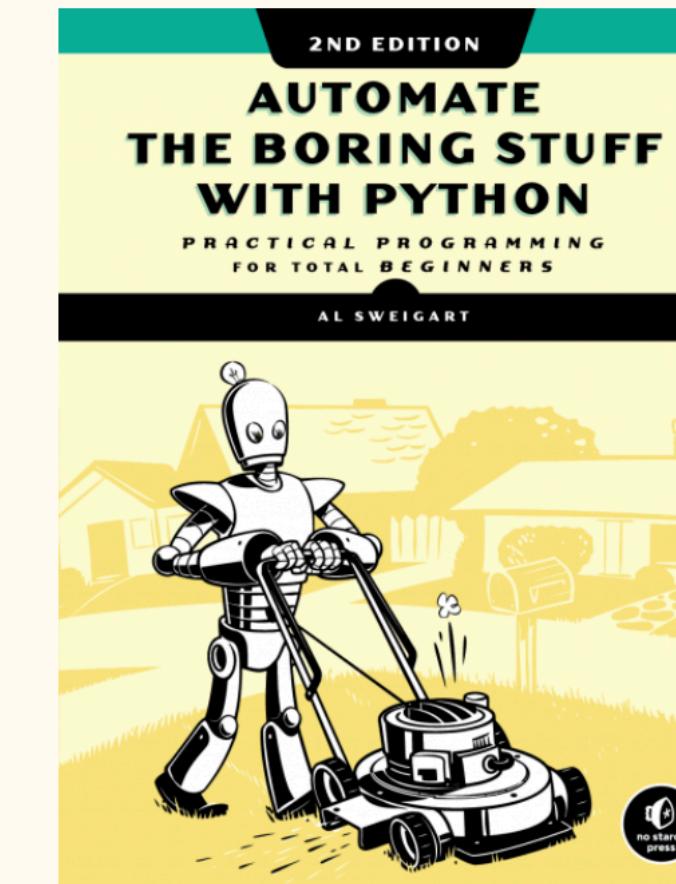
- Books
 - [Think Python](#) ↗
 - [Learn Python the Hard Way](#) ↗ .
 - [Official Python Documentation](#) ↗
 - [WikiBooks Python](#) ↗
- Development environment
 - [Visual Studio Code](#) ↗
 - [Anaconda](#) ↗
 - [GitHub](#) ↗
 - [Python](#) ↗

COURSE RESOURCES

- Automate the Boring Stuff

AUTOMATE THE BORING STUFF WITH PYTHON

By Al Sweigart. Free to read under a [Creative Commons license](#).



Free ebook when bought from [No Starch Press](#)

[Amazon](#)

"The best part of programming is the triumph of seeing the machine do something useful. Automate the Boring Stuff with Python frames all of programming as these small triumphs; it makes the boring fun."

- **Hilary Mason**, Founder of [Fast Forward Labs](#) and Data Scientist in Residence at [Accel](#)

"I'm having a lot of fun breaking things and then putting them back together, and just remembering the joy of turning a set of instructions into something useful and fun, like I did when I was a kid."

- **Wil Wheaton**, [WIL WHEATON dot NET](#)

PRACTICAL PROGRAMMING FOR TOTAL BEGINNERS.

If you've ever spent hours renaming files or updating hundreds of spreadsheet cells,

you'll appreciate how quickly you can get up to speed with the techniques in this book.

COURSE RESOURCES

- Think Python

Think Python – Green Tea Press

Green Tea Press

Free books by Allen Downey

Search ...

GREEN TEA PRESS BOOKS
ALSO PUBLISHED BY
O'REILLY MEDIA

[Think DSP](#)

[Think Java](#)

[Think Bayes](#)

[Think Python 2e](#)

[Think Stats 2e](#)

[Think Complexity](#)

Think Python

Think Python First Edition, by Allen B. Downey

This is the first edition of *Think Python*, which uses Python 2. If you are using Python 3, you might want to use the second edition, [which is here](#).

[You can buy this book at Amazon.com](#)

[Download *Think Python* in PDF.](#)

[Read *Think Python* in HTML.](#)

Example programs and solutions to some problems are [here](#) (links to specific examples are in the book). The code is also available from [this GitHub repository](#).

Description

Think Python is an introduction to Python programming

How to Think Like a Computer Scientist

<https://greenteapress.com/wp/think-python-2e/>

COURSE RESOURCES

- Learn Python the Hard Way



Millions of people have learned to code. So can you.

The original Hard Way book and still the most popular way for total beginners to finally learn how to code. Learn Python The Hard Way takes you from absolute zero to able to read and write basic Python to then understand other books on Python. No experience necessary to begin, and you can even try the book out for free to see if the method works for you. If you've always wanted to learn to code but have no idea where to begin, then this book is for you.

[GET THE BOOK](#)

HTTPS://LEARNPYTHONTHEHARDWAY.ORG/PYTHON3/

COURSE RESOURCES

- Many resources for learning the details of the Python programming language
 - Free digital books available from UChicago library
 - Stack Overflow
 - GitHub
 - The Internet

The screenshot shows the Python.org homepage with a dark blue header. The header includes links for Python, PSF, Docs, PyPI, and Jobs. A search bar with a magnifying glass icon and a 'GO' button is also present. The main content area features the Python logo and navigation tabs for About, Downloads, Documentation, Community, Success Stories, and News. A central box displays Python code for generating a Fibonacci series up to n=1000, with a yellow '...' button next to it. Below this is a quote: "Python is a programming language that lets you work quickly and integrate systems more effectively." At the bottom, there are three columns: 'Get Started' (with a 'Get Started' button and text about learning Python), 'Download' (with a 'Download' button and text about source code and installers), and 'Docs' (with a 'Docs' button and text about documentation). The 'Jobs' section on the right is partially visible.

Welcome to Python.org

python™

About Downloads Documentation Community Success Stories News

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
987
```

...

Python is a programming language that lets you work quickly and integrate systems more effectively. [»» Learn More](#)

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

[Start with our Beginner's Guide](#)

Download

Python source code and installers are available for download for all versions! Not sure which version to use? [Check here](#).

Latest: [Python 3.6.0](#) - [Python 2.7.13](#)

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

[docs.python.org](#)

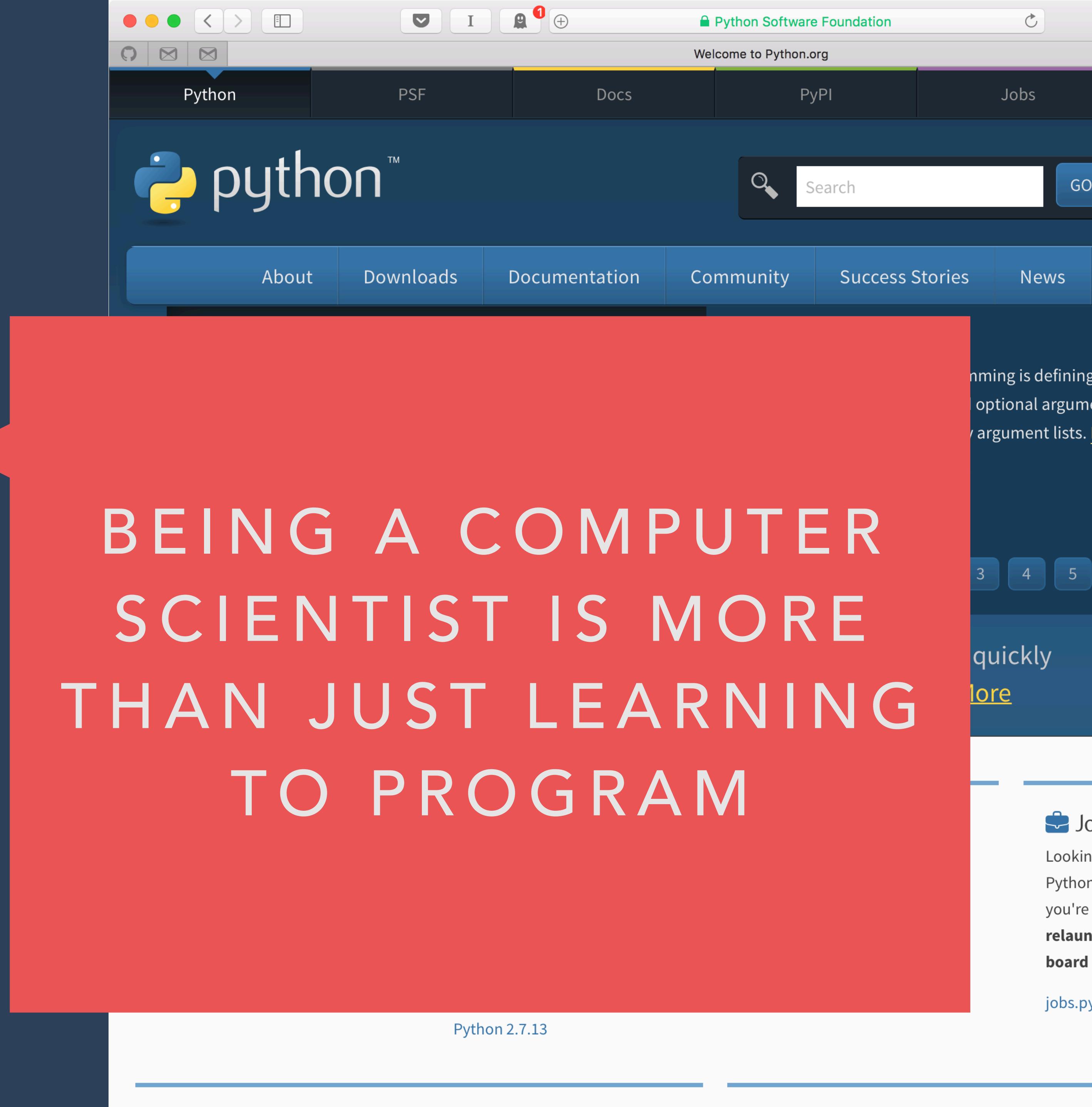
Jobs

Looking for Python jobs? [Relaunch board](#)

[jobs.py](#)

COURSE RESOURCES

- Many resources for learning the details of the Python programming language
 - Free digital books available from UChicago library
 - Stack Overflow
 - GitHub
 - The Internet



BREAKOUT EXERCISES

© T.A. BINKOWSKI, 2020

MODULE 1
MPCS 50101



THE UNIVERSITY OF
CHICAGO

BREAKOUT EXERCISES

Module 1: Breakout Exercises

Work together to complete the exercises below. You may be asked to present your answer to the class.

Exercise 1

Once upon a time a farmer went to a market and purchased a wolf, a goat, and a cabbage. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and a single one of his purchases: the wolf, the goat, or the cabbage.



If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage.

The farmer's challenge was to carry himself and his purchases to the far bank of the river, leaving each purchase intact.

MODERN COMPUTING AND COMPUTATIONAL THINKING

© T.A. BINKOWSKI, 2020

MODULE 1
MPCS 50101



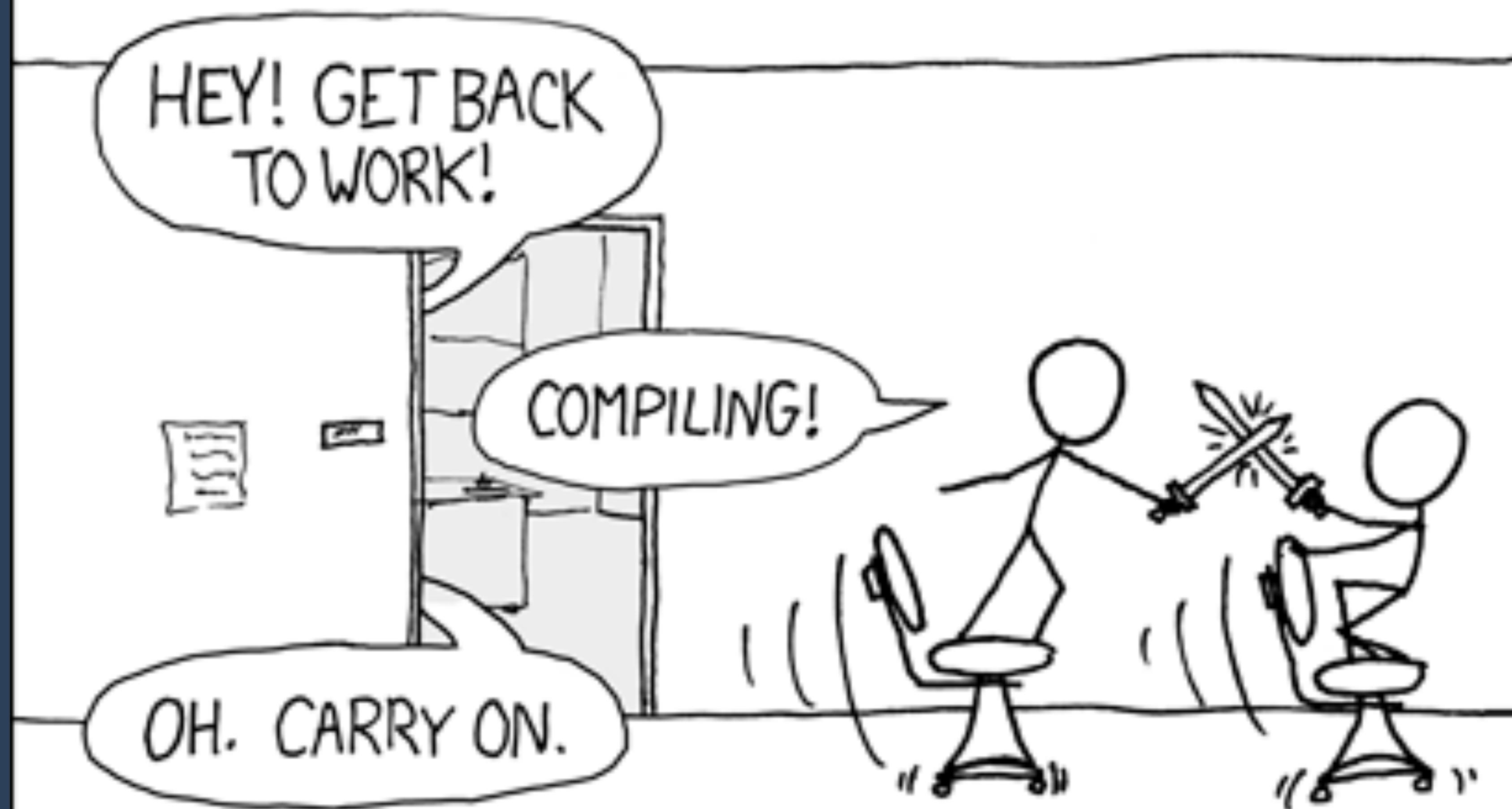
THE UNIVERSITY OF
CHICAGO

MODERN COMPUTING

- What does a computer scientist do?

THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:

"MY CODE'S COMPILING."



MODERN COMPUTING

- Computer scientists think.
 - How to solve a problem
 - How to solve a problem using a computer

THESE CAN BE VERY
DIFFERENT



MODERN COMPUTING

- Being a computer scientist has less to do with coding than thinking
- Problem solving
 - Many times the problems you create for yourself



**EVERYTHING YOU
NEED TO KNOW
ABOUT
PROGRAMMING**

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- A program is just a set of instructions how to perform a computation
- Examples of programs
 - Format text
 - Find the area of a complex shape
 - Target ads to kids
 - Detect objects in an image



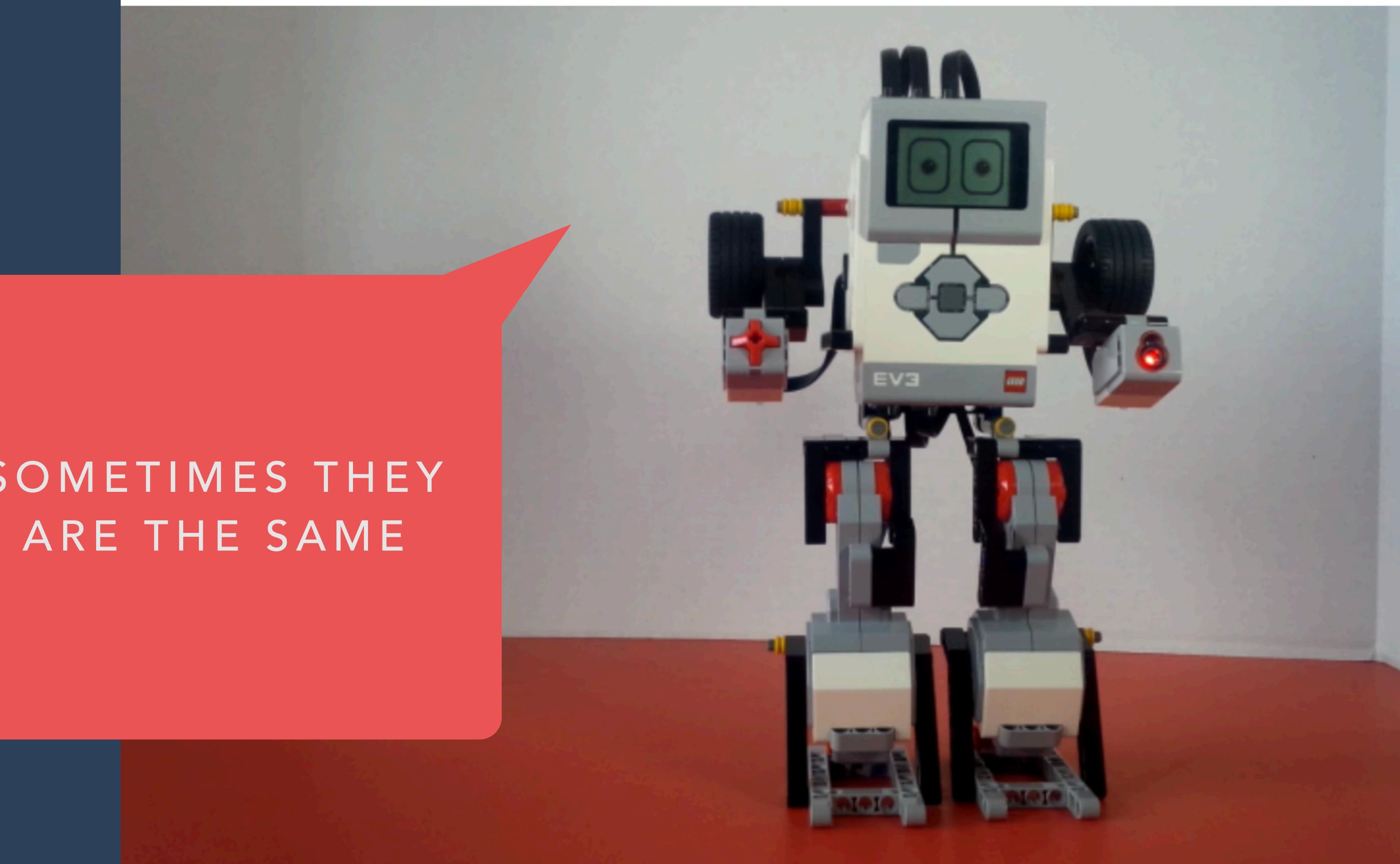
THEY'RE ALL JUST
FOLLOWING
INSTRUCTIONS

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- Programs for humans
 - Dance, bake a cake
 - Drive a car
- Programs for computers
 - Dance, bake a cake
 - Drive a car

Lego Mindstorms Dancing Robot

12/05/2015



SOMETIMES
THEY
ARE THE SAME

This week our family project with the kids was to build and program a Lego Mindstorms Dancing Robot. I thought this would be a great exercise in computer programming for the kids because a computer program is just a sequence of instructions, and a dance routine is a series of movements. Programming a robot to make specific movements in a set order seemed like a great way for them to get an immediate visual reward for their efforts.

EVERYTHING YOU NEED TO
KNOW ABOUT
PROGRAMMING

SOMETIMES
NOT

Tesla Bears Some Blame for Self-Driving Crash Death, Feds Say



EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING



A HUMAN DOING THE ROBOT
OR
A ROBOT DOING THE HUMAN

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- To get the desired results programs for both humans and computers require

PRECISION and
ACCURACY



**COMPONENTS OF
EVERY COMPUTER
PROGRAM**

COMPONENTS OF EVERY COMPUTER PROGRAM

YOU ALREADY KNOW
HOW TO DO THIS

```
dogs = 4  
cats = 2
```

```
if dogs > cats:  
    print("I love dogs more")
```

```
total_animals = dogs + cats  
print("I have %d animals" % total_animals)
```

COMPONENTS OF EVERY COMPUTER PROGRAM

```
if 'hello' in ['hello', 'goodbye', 'farewell']:  
    print('We found hello!')
```

```
if 10 > 20:  
    print('10 is greater than 20')
```

```
print('B')
```

```
if 10 < 20:  
    print('10 is less than 20')
```

```
print('Done!')
```

COMPONENTS OF EVERY COMPUTER PROGRAM

```
if 'hello' in ['hello', 'goodbye', 'farewell']:  
    print('We found hello!')
```

```
if 10 > 20:  
    print('10 is greater than 20')
```

```
    print('B')
```

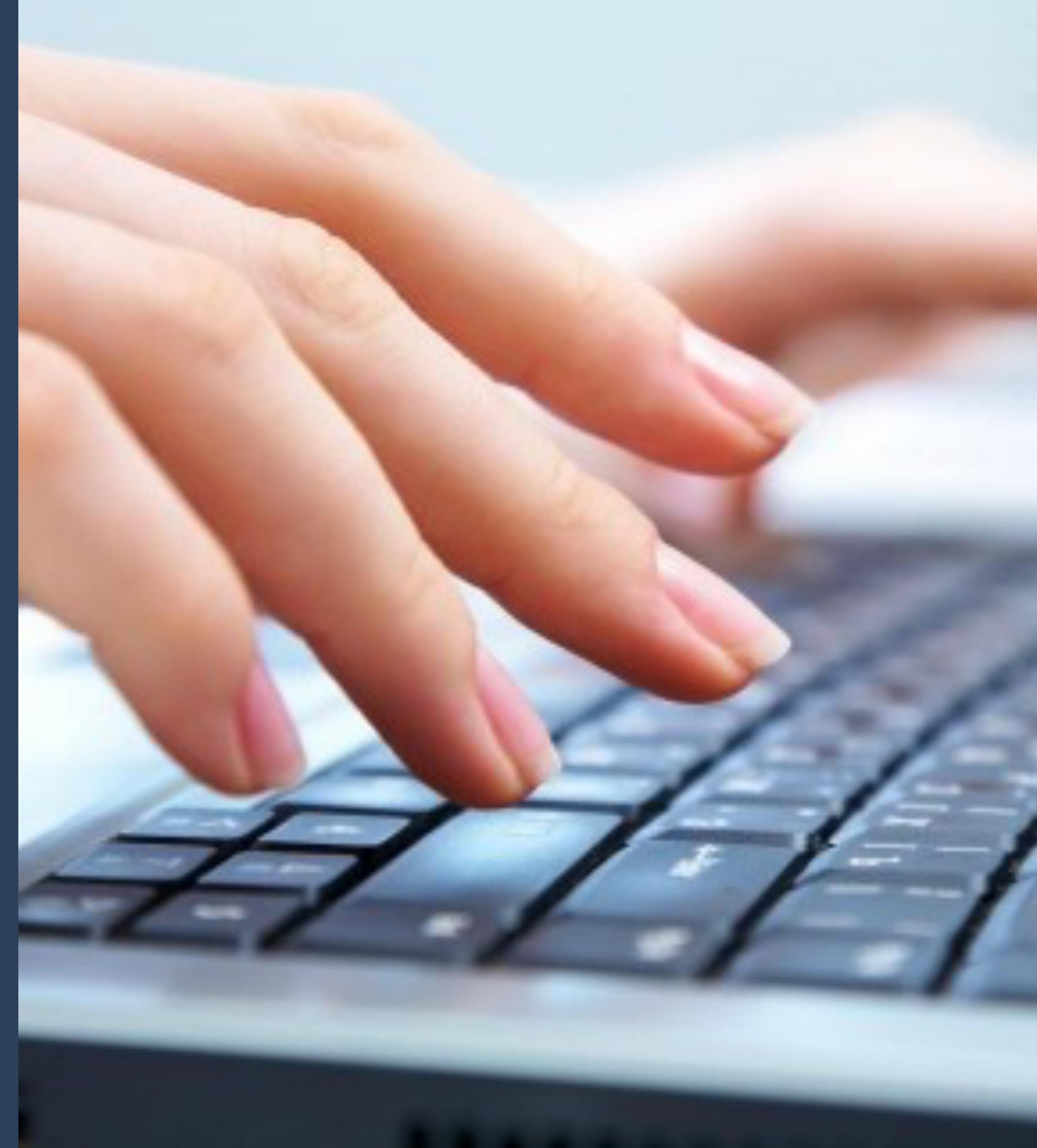
```
if 10 < 20:  
    print('10 is less than 20')
```

```
print('Done!')
```

WE FOUND HELLO!
B
10 IS LESS THAN 20
DONE!

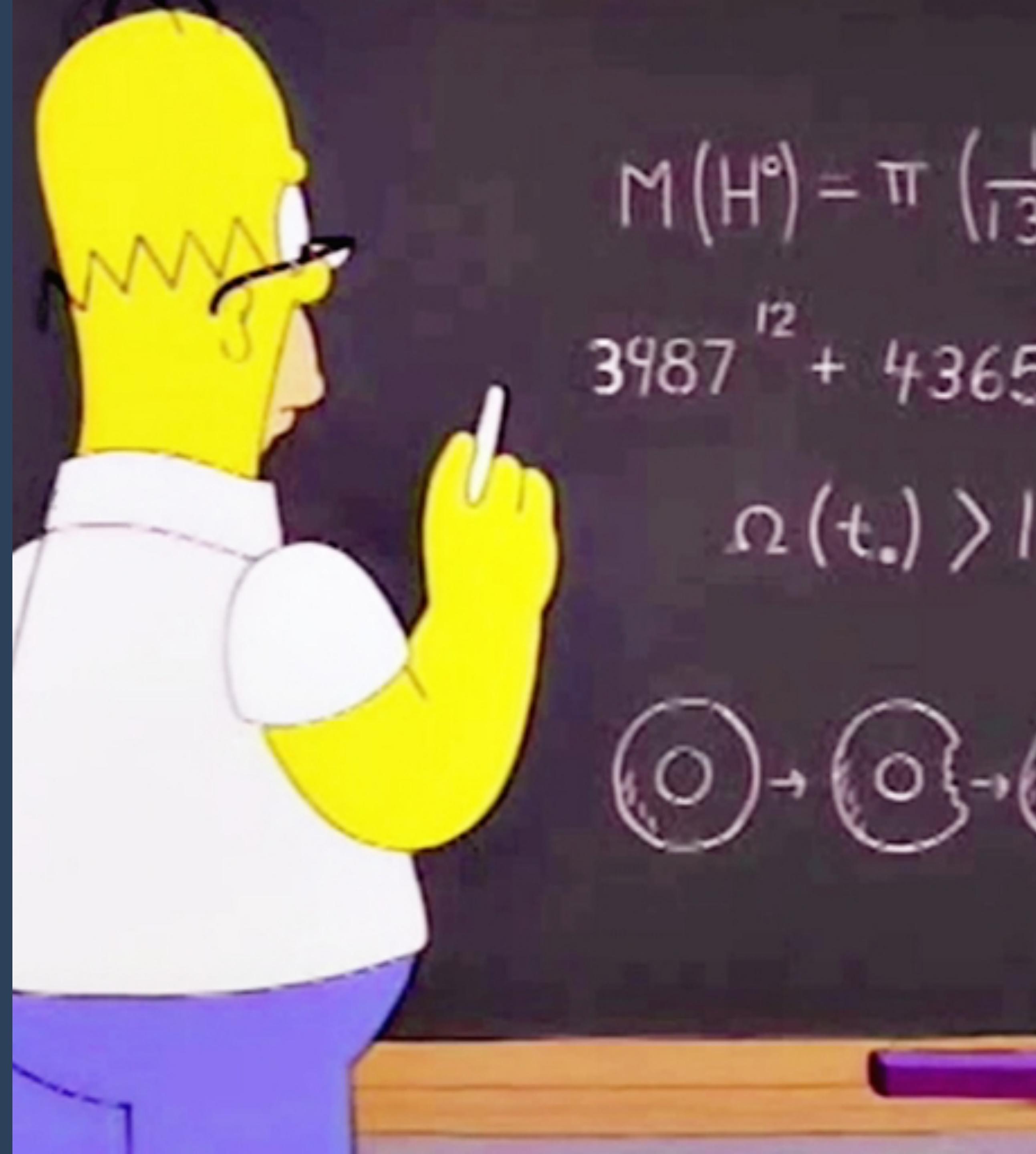
COMPONENTS OF EVERY COMPUTER PROGRAM

- Input
 - Data from keyboard touches
 - A file
 - Database
 - Sensors
 - ...



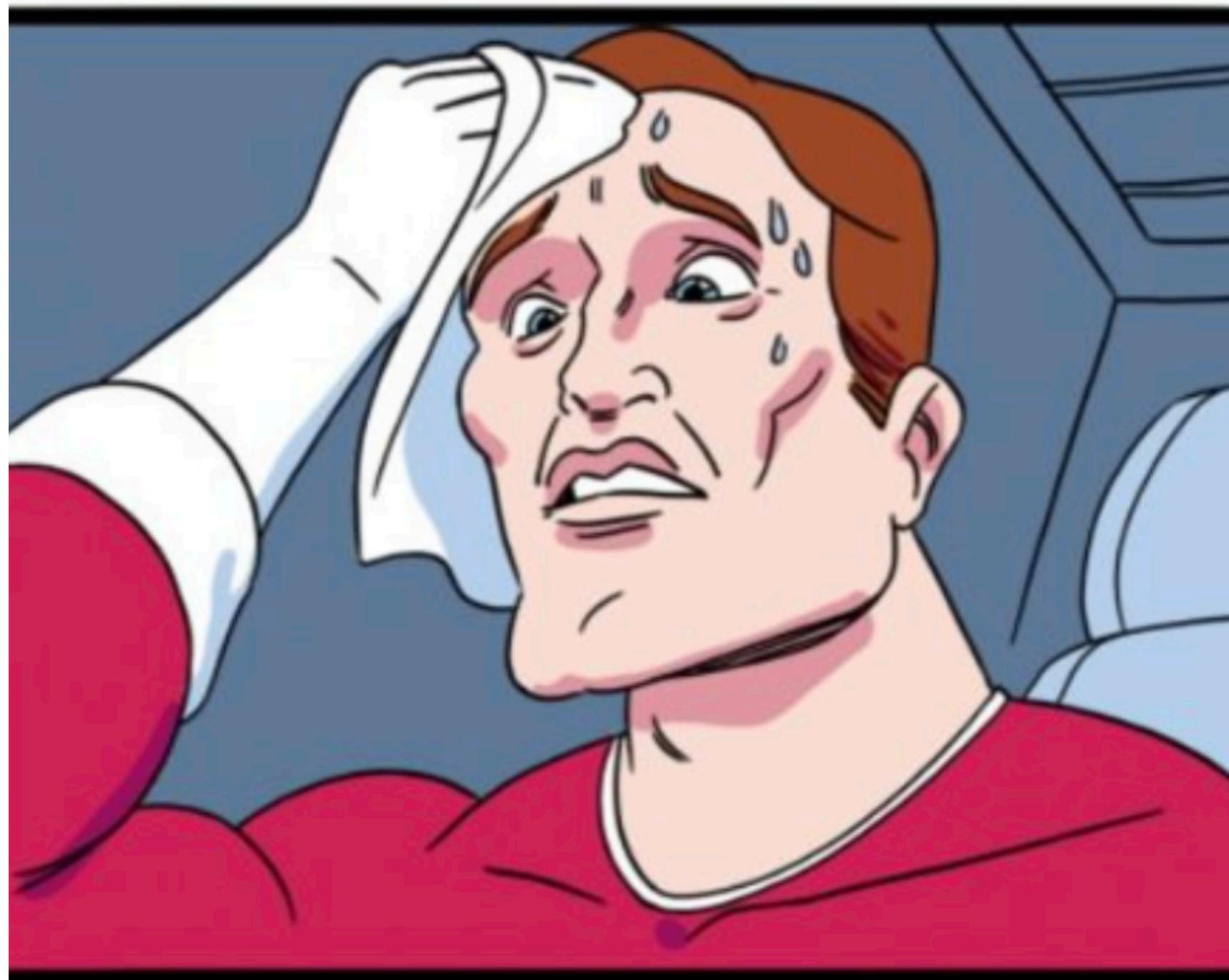
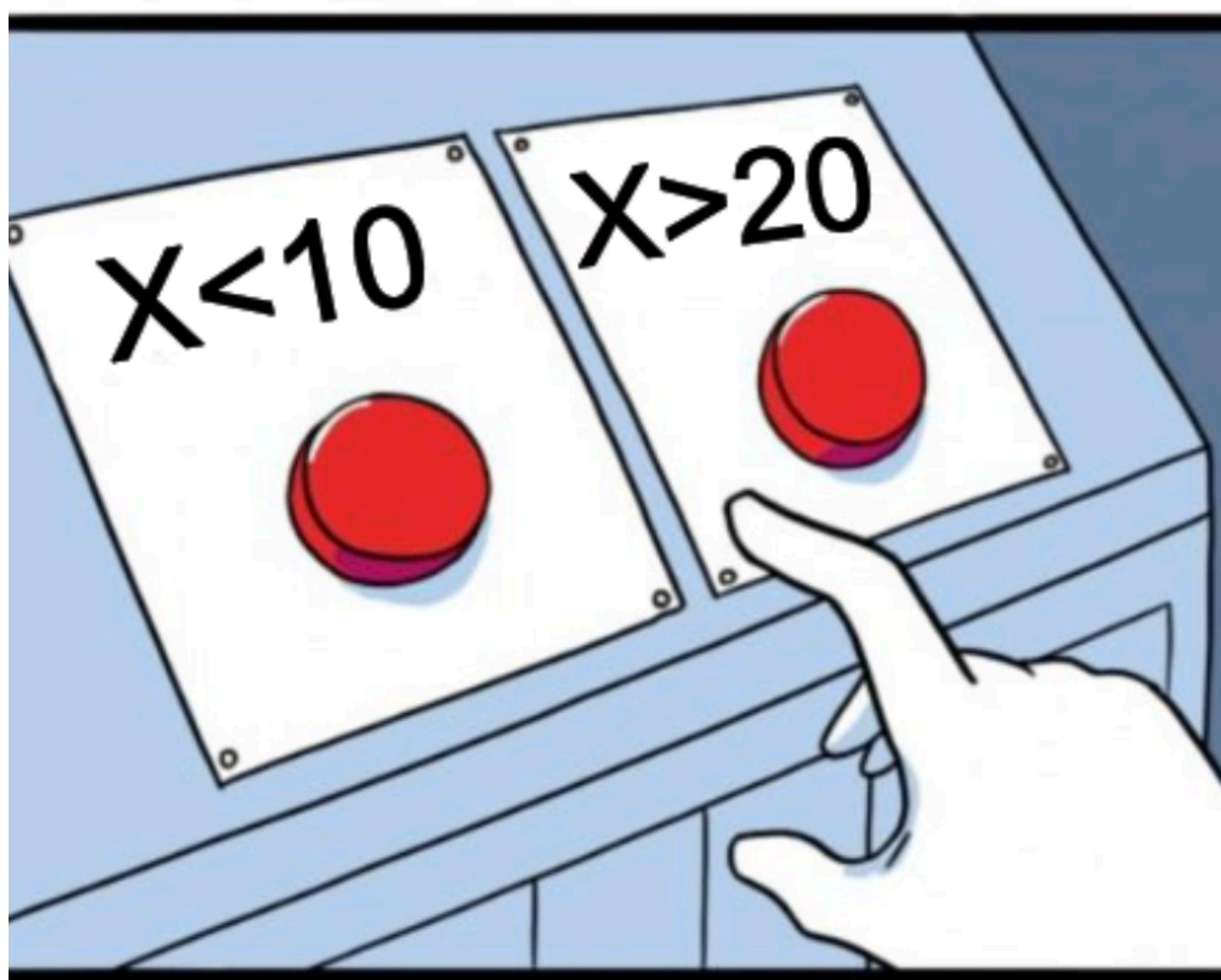
COMPONENTS OF EVERY COMPUTER PROGRAM

- Math
 - Mathematical operations
 - Logic operations
 - Behind the scenes operations



COMPONENTS OF EVERY COMPUTER PROGRAM

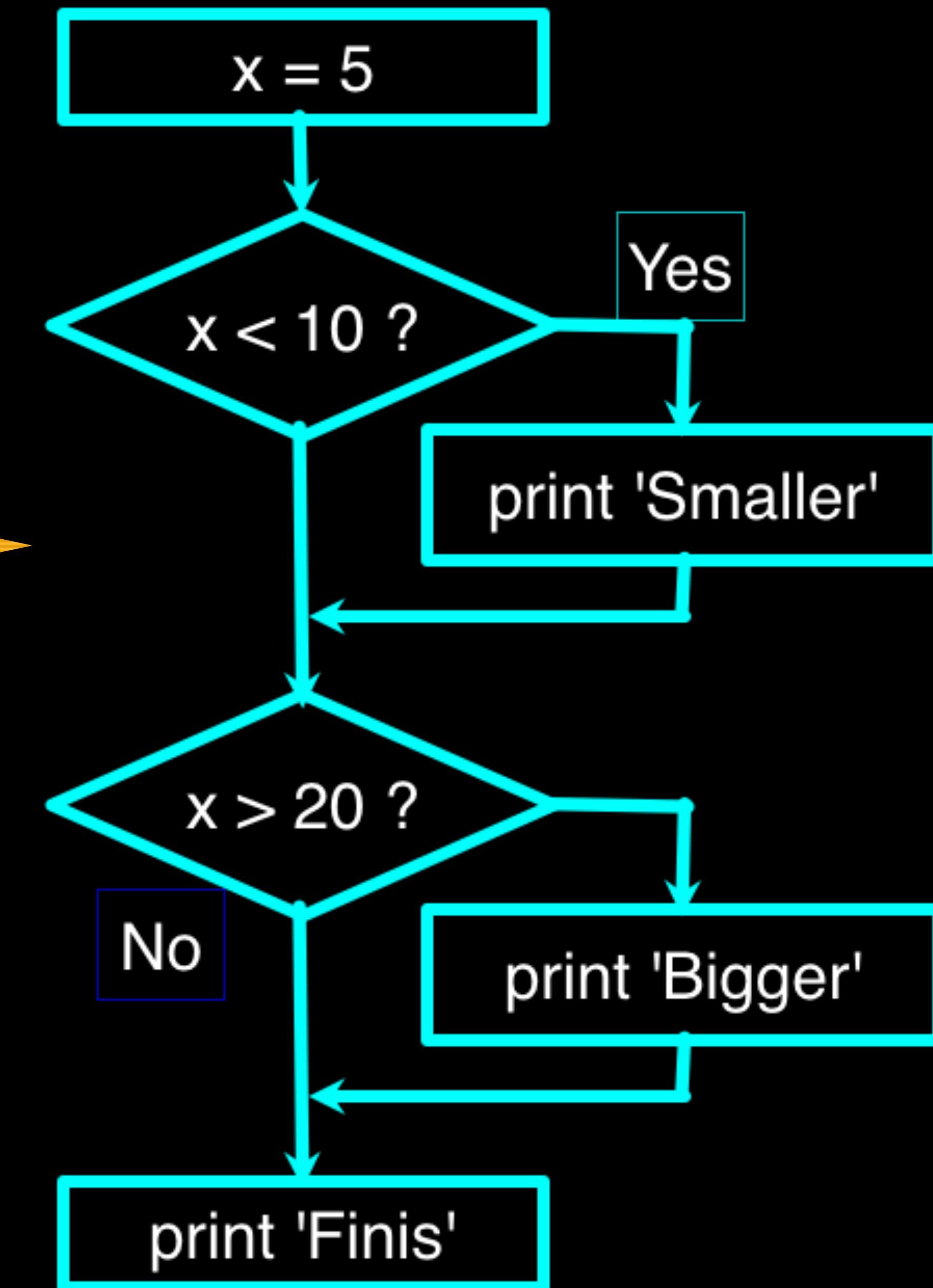
- Conditional Execution
 - Check for conditions and execute appropriate code
 - Some code paths will never be executed 😳



COMPONENTS OF EVERY COMPUTER PROGRAM

- Conditional Execution
- Check for conditions and execute appropriate code
- Some code paths will never be executed at runtime

PROBABLY SOMETHING MORE LIKE THIS

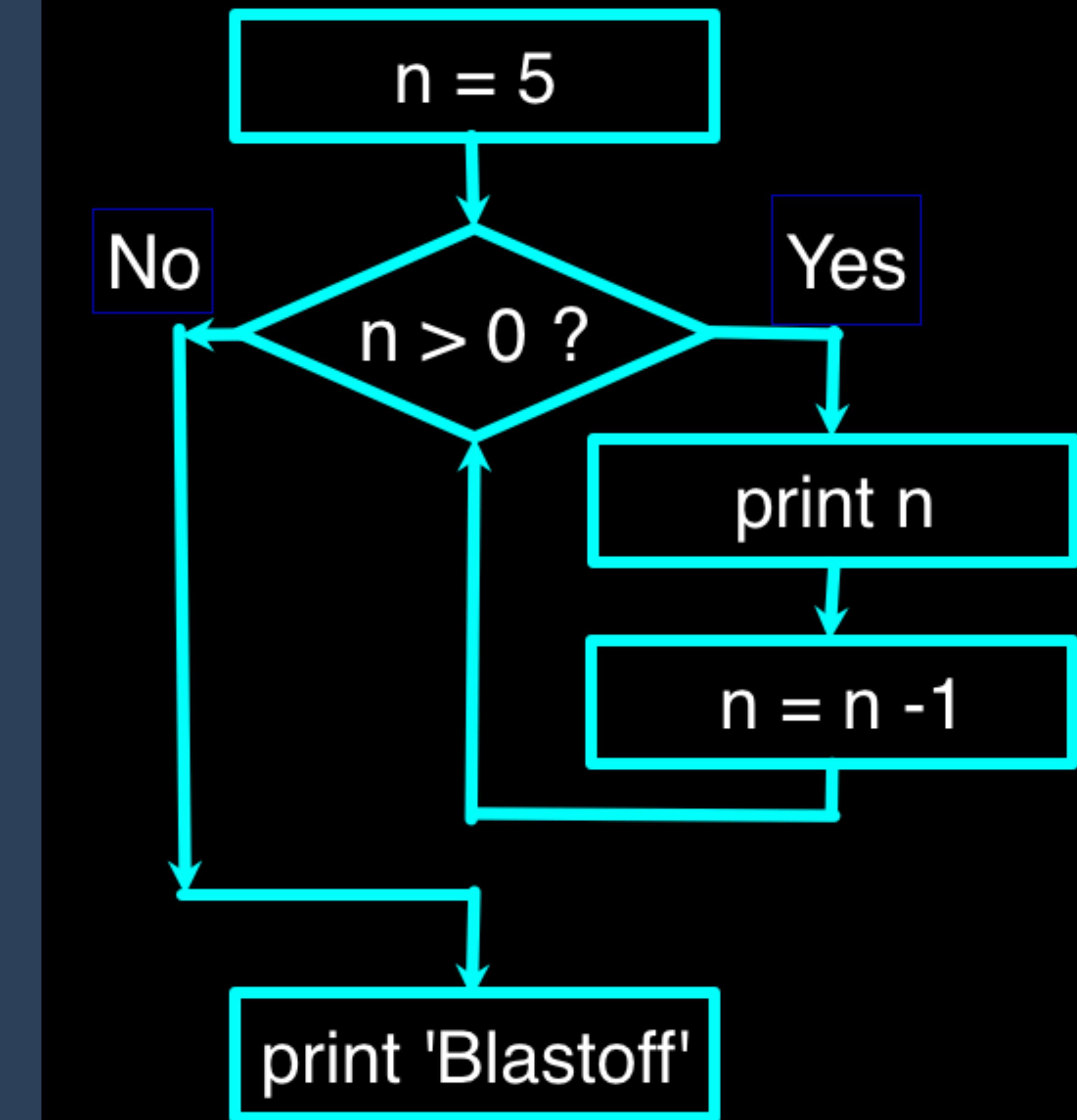


COMPONENTS OF EVERY COMPUTER PROGRAM

```
if x < 10:  
    print("smaller")  
  
if x > 20:  
    print("bigger")  
  
print("finished")
```

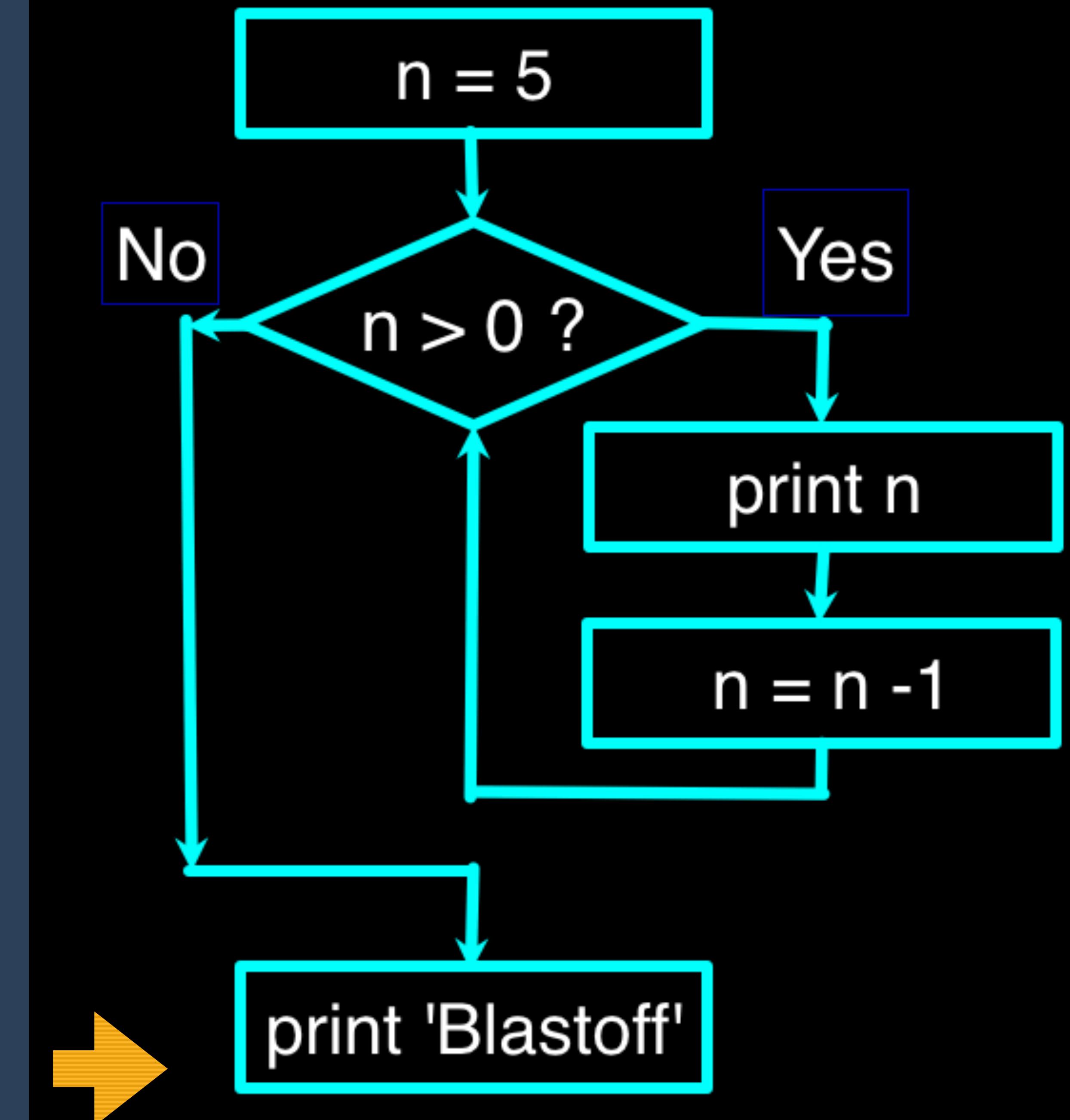
COMPONENTS OF EVERY COMPUTER PROGRAM

- Repetition
 - Perform same action repeatedly
 - Often combined with conditional statements



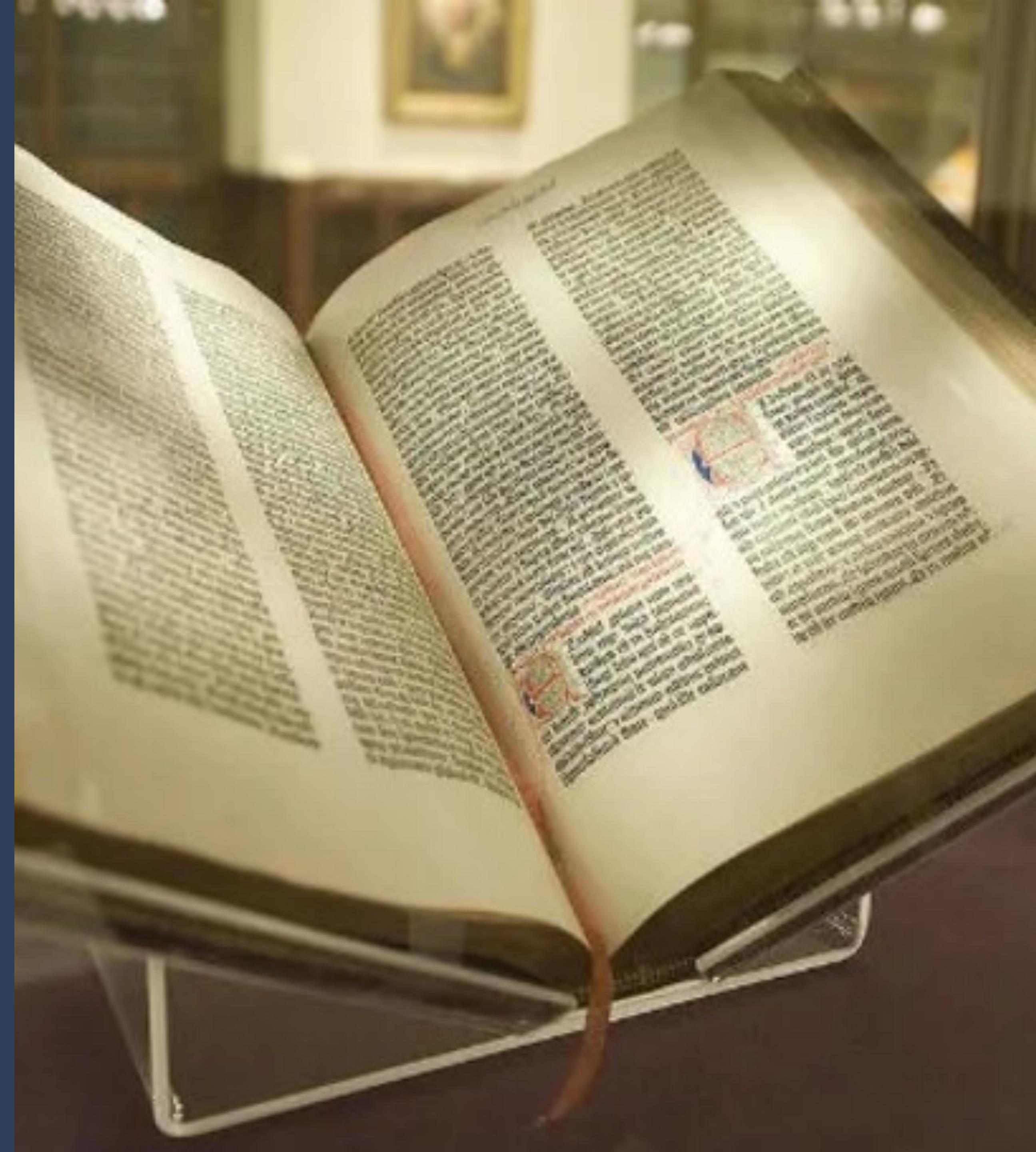
COMPONENTS OF EVERY COMPUTER PROGRAM

- Output
 - Display data on the screen
 - Save to file
 - ...



EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- Let's write a program to
“Count Words in a Book”



EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

COUNT WORDS
IN A BOOK

select a book

open the book

count every word on the page

if there is another page, turn it
count every word on the page

If there are no more pages,
Print how many words we saw

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

COUNT WORDS
IN A BOOK

select a book

INPUT

open the book

REPETITION

count every word on the page

CONDITIONAL

if there is another page, turn it
count every word on the page

REPETITION

If there are no more pages,
Print how many words we saw

OUTPUT

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

```
name = raw_input('Enter file: ')
handle = open(name, 'r')
text = handle.read()
words = text.split()
```

INPUT

COUNT WORDS
IN A FILE

```
counts = dict()
for word in words:
    counts[word] = counts.get(word, 0) + 1
```

REPETITION

```
bigcount = None
bigword = None
```

REPETITION

```
for word, count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count
```

CONDITIONAL

```
print bigword, bigcount
```

OUTPUT

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

select a book

open the book

count every word on

if there is another
count every word on

If there are no more
say how many times

PRECISION AND ACCURACY

- IS THERE A BOOK?
- IS IT IN A LANGUAGE WE KNOW ABOUT?
- IS THERE ANOTHER PAGE?
- IS THE PAGE BLANK? DOES THAT MEAN ITS THE END OF THE BOOK?
- WHAT IF THERE ARE CONSECUTIVE BLANK PAGES?
- DOES THE COVER COUNT?

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

select a book

open the book

count every word on

if there is another
count every word on

If there are no more
say how many times

PRECISION AND ACCURACY

- IS THERE A BOOK?
- IS IT IN A LANGUAGE WE KNOW ABOUT?
- IS THIS THE ITERATIVE PART OF WRITING PROGRAMS...STUFF YOU DIDN'T THINK OF
- IS THE PAGE BLANK? THE FIRST TIMES THAT MEAN
(EDGE CASES, BUGS, ETC.)
- IS IT THE END OF THE BOOK?
- WHAT IF THERE ARE CONSECUTIVE BLANK PAGES?
- DOES THE COVER COUNT?

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- The language and platform may differ, but that's it 😊
- For any given problem
 - Break it down in to these components
 - Write it in a way a computer can understand

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- There are many ways to write a program to solve a problem
- How do we evaluate "good" programming
 - Efficiency
 - Readability
 - Style

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- There are many ways to write a program to solve a problem
- How do we evaluate "good" programming

- Efficiency



OBJECTIVE

- Readability



HIGHLY
SUBJECTIVE

- Style

EVERYTHING YOU NEED TO KNOW ABOUT PROGRAMMING

- There are many ways to write a program.
- How do we evaluate "good" programs?
 - Efficiency
 - Readability
 - Style

HIGHLY
SUBJECTIVE

HEART TO HEART TALK

YOU WILL ENCOUNTER A LOT OF NEGATIVITY,
SCARCASM, ELITISM AND SNARK
WHEN LEARNING COMPUTER SCIENCE.

IGNORE IT.

THERE ARE ENOUGH PEOPLE WHO
ARE KIND, ETHUSIASTIC AND
GENUINELY EXCITED TO HELP.
FIND THOSE PEOPLE.

MODULE OVERVIEW

MODULE 1
MPCS 50101

© T.A. BINKOWSKI, 2020



THE UNIVERSITY OF
CHICAGO

MODULE OVERVIEW

- Module 1

- ⋮ ▾ **Module 1: Welcome to Computing**
- ⋮  **Module 1: Overview**
- ⋮  **Resources**
- ⋮  **Module 1: Breakout Exercises**
- ⋮  **Module 1: Lecture Slides**
- ⋮  **Lecture Videos**
- ⋮  **Module 1: Welcome to Programming**
- ⋮  **Module 1: Setting Up A Development Environment**
- ⋮  **Assignments**
- ⋮  **Module 1: Introduction Survey**
Oct 4
- ⋮  **Module 1: Lecture Quiz**
Oct 6 | 7 pts
- ⋮  **Module 1: Assignment**
Oct 6 | 10 pts

THE END

MPCS 50101
MODULE 1



THE UNIVERSITY OF
CHICAGO

©TTAA .BB INKKOWSKI ,22019