CSC-121

Introduction to Computer Programming

Lecture 0

About Me

Name: Syed Fahad Sultan سید فہد سلطان

Pronunciation (IPA): ˈsæjjɪd fah(aː)d solˈtˤɑːn

	First name	Middle name	Last name
Syntax	Syed	Fahad	Sultan
Semantics	Fahad (given name)	Sultan (middle name)	Syed (family name)

Just call me "Dr. Sultan" (Pronounced: Sool TAHN)

Office: Riley Hall 200-D

Email: fahad.sultan@furman.edu

Phone: 864-294-3755

Office hours:

Monday: 1:30 PM – 4:30 PM
Thursday: 9:30 AM – 12:30 PM

Email for appointment
Open door policy, when not in class or meeting

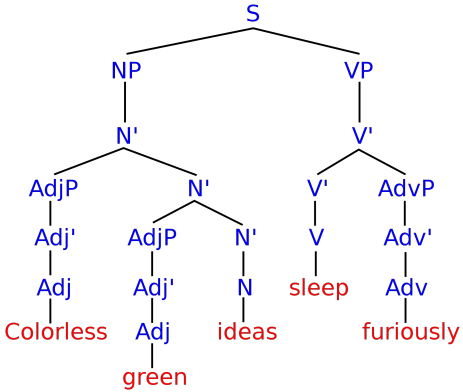
About this course

- Learning computer programming is a lot like **learning a new language**
  - Different programming languages use different syntax (grammar)
  - In this course, we'll learn to **read and write Python**

Programming Language	**Syntax (grammar)**	**Semantics (meaning/logic)**
Python	`print("Hello world")`	Print "*Hello world*"
Java	`System.out.println("Hello world")`	Print "*Hello world*"
C++	`std::cout<<"Hello world";`	Print "*Hello world*"
Javascript	`console.log("Hello world")`	Print "*Hello world*"
PHP	`echo "Hello world";`	Print "*Hello world*"

About this course

- **Syntax** (grammar) for any language (programming or not) may be learned passively from slides or a book
    - There is no textbook for this course
- In this course, you'll be **graded *mostly* on semantics** (logic)
- Correct syntax with incorrect logic will ***not*** get you points
- For assignments, labs and any other tests/exams on computers, syntax errors will cost you heavily: at least 60% of the assignment grade
    - Don't worry. Syntax errors are easy to catch and correct in a programming environment
- For exams/tests on paper, correct logic with incorrect syntax, is acceptable, *within reason*




```
>>> if 2 == 3:
    print('hello')
else if 2 == 2:
    print('goodbye')

SyntaxError: invalid syntax
```

Practice, Practice, Practice!

- The **only one way** to get good at reading and writing any language:
    - Lots and lots of **Practice**!
- You will have to learn to **express your ideas in code** to a computer (and to me)
- You will be given plenty of opportunities to practice:
- In-class problems (10% of the total course grade)
    - We'll work through these together
    - You are expected to submit on Moodle
- Weekly assignments (20% of the total course grade)
- Weekly labs (10% of the course grade)
    - You will work individually, with minimal assistance from me

Class Participation

- ***Please bring your laptops to every class and lab session***
    - Please let me know if you don't have access to a laptop
- General flow of a class
    1. I will introduce a new programming/Python concept
    2. I will pose problem(s) relevant to the concept
    3. We will work together to solve the problem(s)
    4. You will submit your work to Moodle
- Class Participation grade (10%) depends on
    - Participation in the collaborative problem solving
    - Submission of your work
- In the first lab, we'll go over the programming environment
    - Anaconda
    - Jupyter Notebook
    - Sublime

Topic Week

Learning

- A new topic every Friday
- Three lectures of guided collaborative problem solving
- Office hours

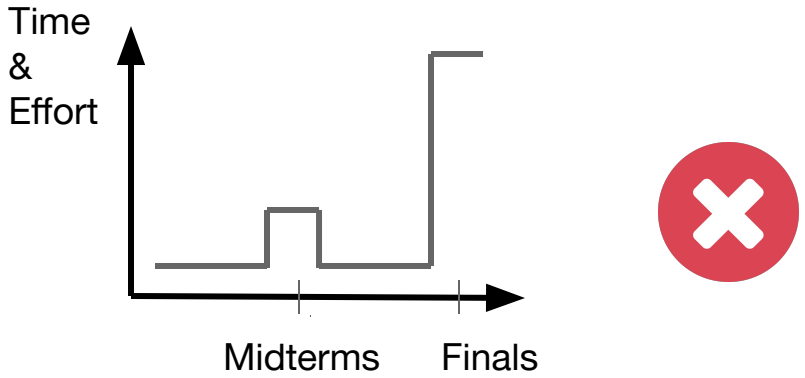
Assessments

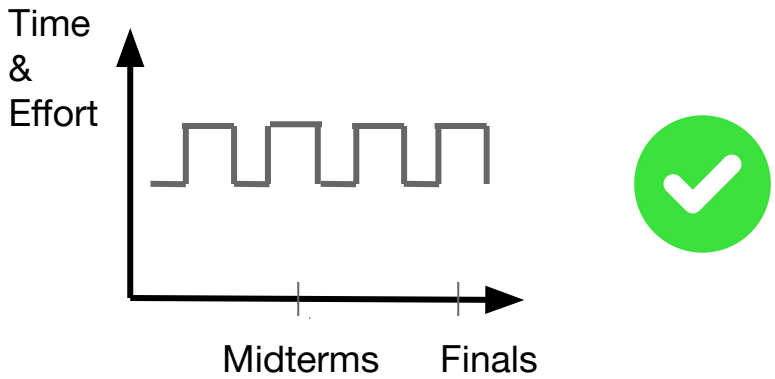
- Towards the end of the *topic week*
- Weekly labs (Wednesdays/Thursdays)
- Weekly assignments, due Thursday, before midnight

	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday
8:30 - 9:30	**Assignment posted**						
9:30 - 10:30							Office Hours (Riley 200-D)
10:30 - 11:30	121-01 (Riley 106)			121-01 (Riley 106)		CSC121-01 (Riley 106)	
11:30 - 12:30	121-02 (Riley 106)			121-02 (Riley 106)		CSC121-02 (Riley 106)	
12:30 - 1:30							
1:30 - 2:30				Office Hours (Riley 200-D)			
2:30 - 3:30						**121-01 Lab (Riley 201)**	**121-02 Lab (Riley 201)**
3:30 - 4:30							
4:30 - 5:30							**Assignment due (11:59 PM)**

Grade Breakdown

Assignments	20
Class Participation	10
Labs	10
Midterm 1 Friday, September 16th	**15**
Midterm 2 Friday, October 14th	**15**
Midterm 3 Friday, October 28th	**15**
Final Monday, December 12th	**15**

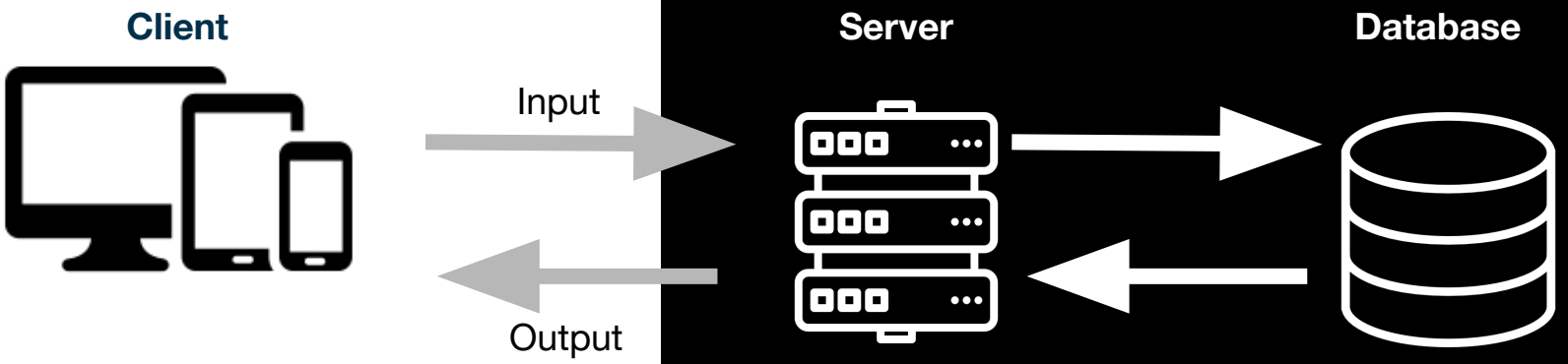




Anatomy of an Application

FRONT END (GRAPHICAL USER INTERFACE)

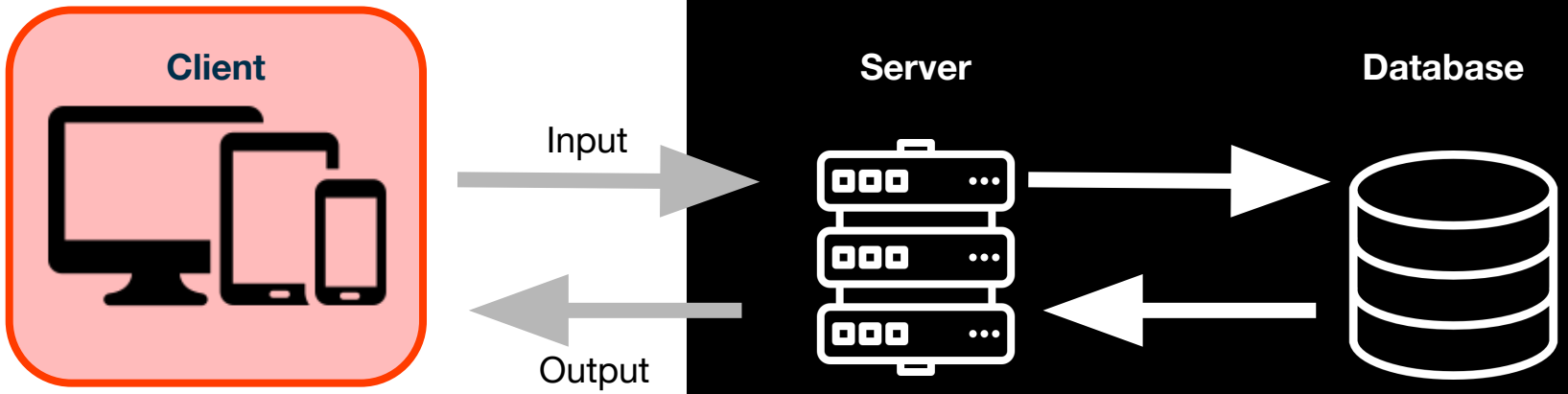
BACK END (COMMAND LINE/TERMINAL)



Anatomy of an Application

FRONT END (GRAPHICAL USER INTERFACE)

BACK END (COMMAND LINE/TERMINAL)

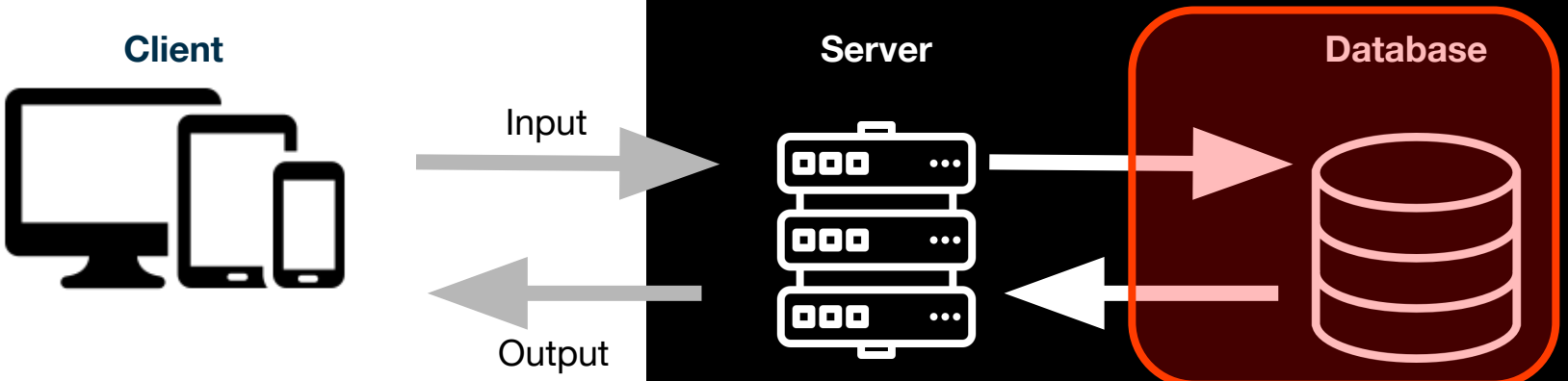


CSC-241 Mobile Apps (*Objective-C, Java*)
CSC-342 Web-Based Apps (*HTML, Javascript*)
CSC-353 Software Engineering
CSC-347 Human Computer Interaction

Anatomy of an Application

FRONT END (GRAPHICAL USER INTERFACE)

BACK END (COMMAND LINE/TERMINAL)



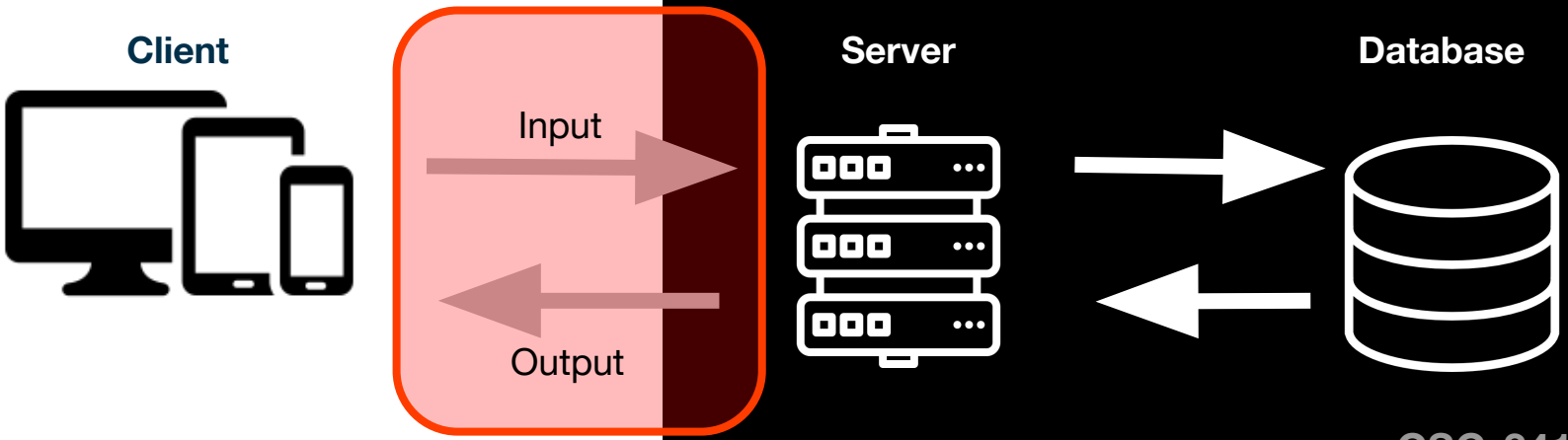
CSC-241 Mobile Apps (*Objective-C, Java*)
CSC-342 Web-Based Apps (*HTML, Javascript*)
CSC-353 Software Engineering
CSC-347 Human Computer Interaction

CSC-341 Database Management Systems (SQL)

Anatomy of an Application

FRONT END (GRAPHICAL USER INTERFACE)

BACK END (COMMAND LINE/TERMINAL)



CSC-241 Mobile Apps (*Objective-C, Java*)
CSC-342 Web-Based Apps (*HTML, Javascript*)
CSC-353 Software Engineering
CSC-347 Human Computer Interaction

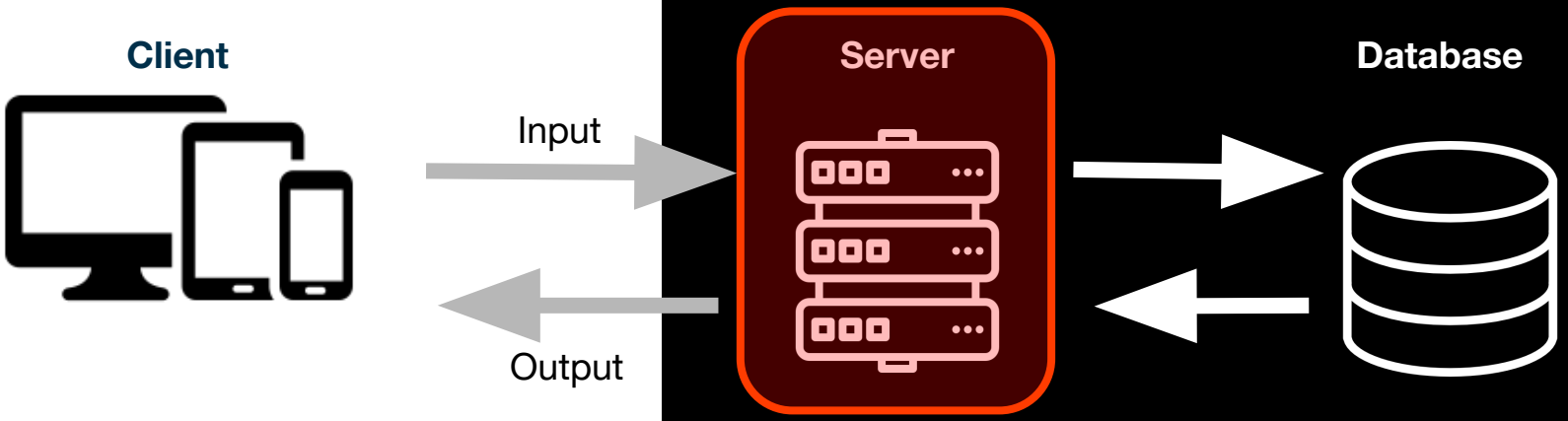
CSC-332 Data Communications and Networking

CSC-341 Database Management Systems

Anatomy of an Application

FRONT END (GRAPHICAL USER INTERFACE)

BACK END (COMMAND LINE)

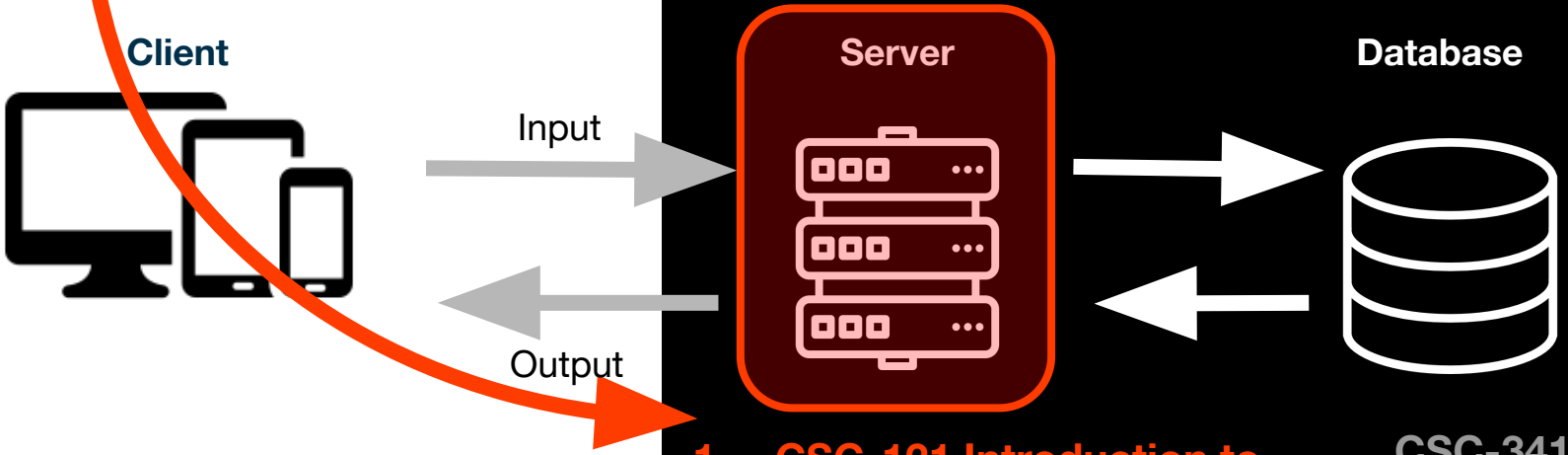


CSC-241 Mobile Apps (*Objective-C, Java*)
CSC-342 Web-Based Apps (*HTML, Javascript*)
CSC-353 Software Engineering
CSC-347 Human Computer Interaction
CSC-332 Data Comm. and Networking

1. CSC-121 Introduction to Programming
2. CSC-122 Data Structures and Algorithms
3. CSC-223 Advanced Data Structures & Algos.

CSC-341 Database Management Systems

You are here

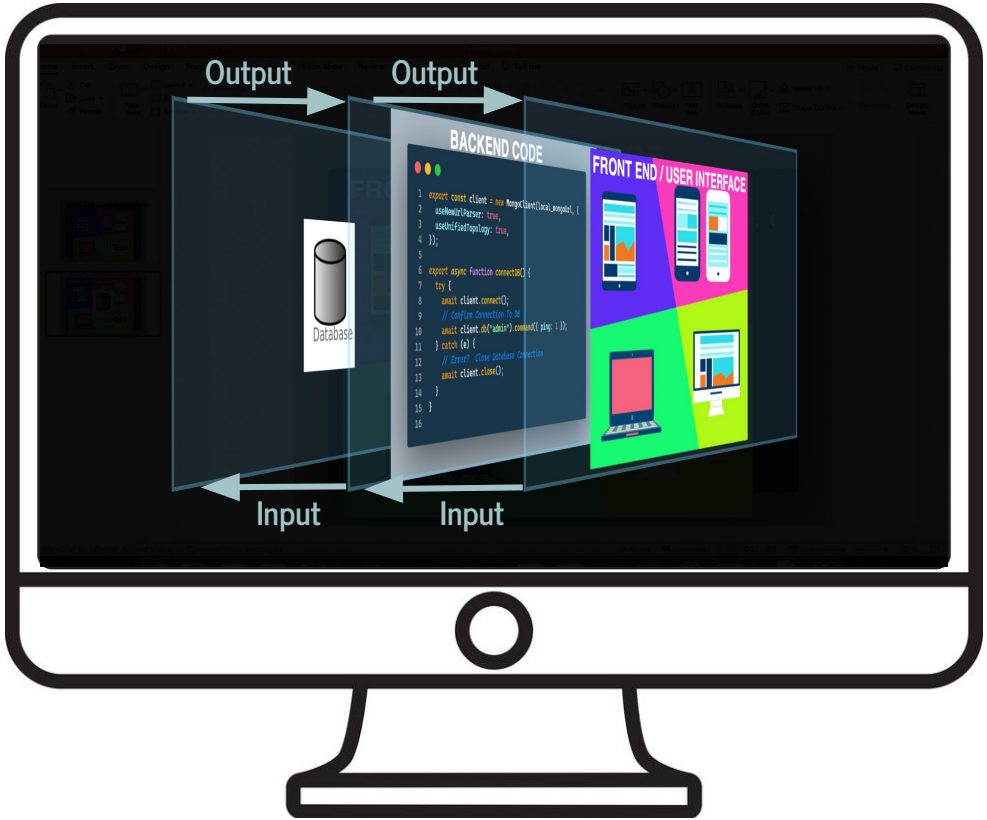


CSC-241 Mobile Apps (*Objective-C, Java*)
CSC-342 Web-Based Apps (*HTML, Javascript*)
CSC-353 Software Engineering
CSC-347 Human Computer Interaction
CSC-332 Data Comm. and Networking

1. CSC-121 Introduction to Programming
2. CSC-122 Data Structures and Algorithms
3. CSC-223 Advanced Data Structures & Algos.

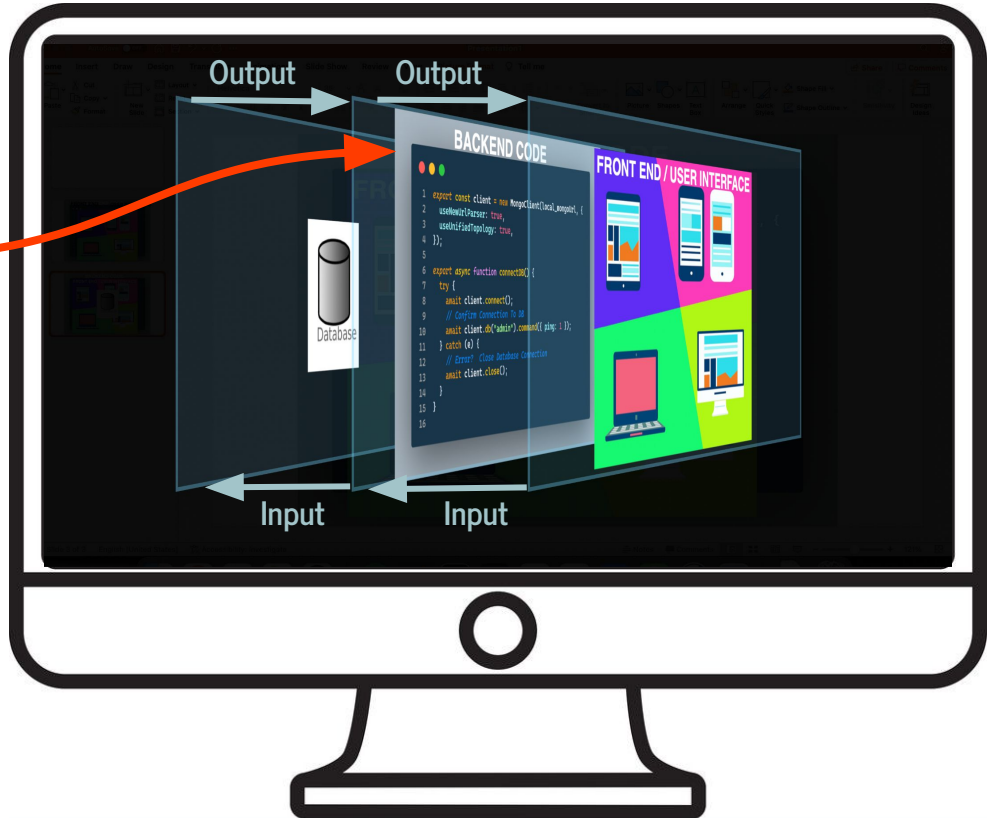
CSC-341 Database Management Systems

Anatomy of (Desktop) Applications



Anatomy of (Desktop) Applications

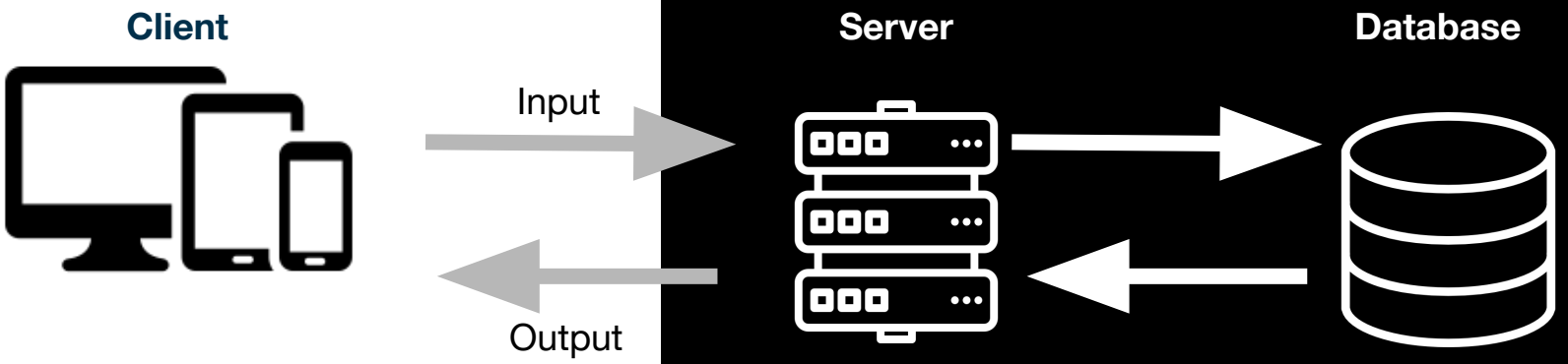
You are here



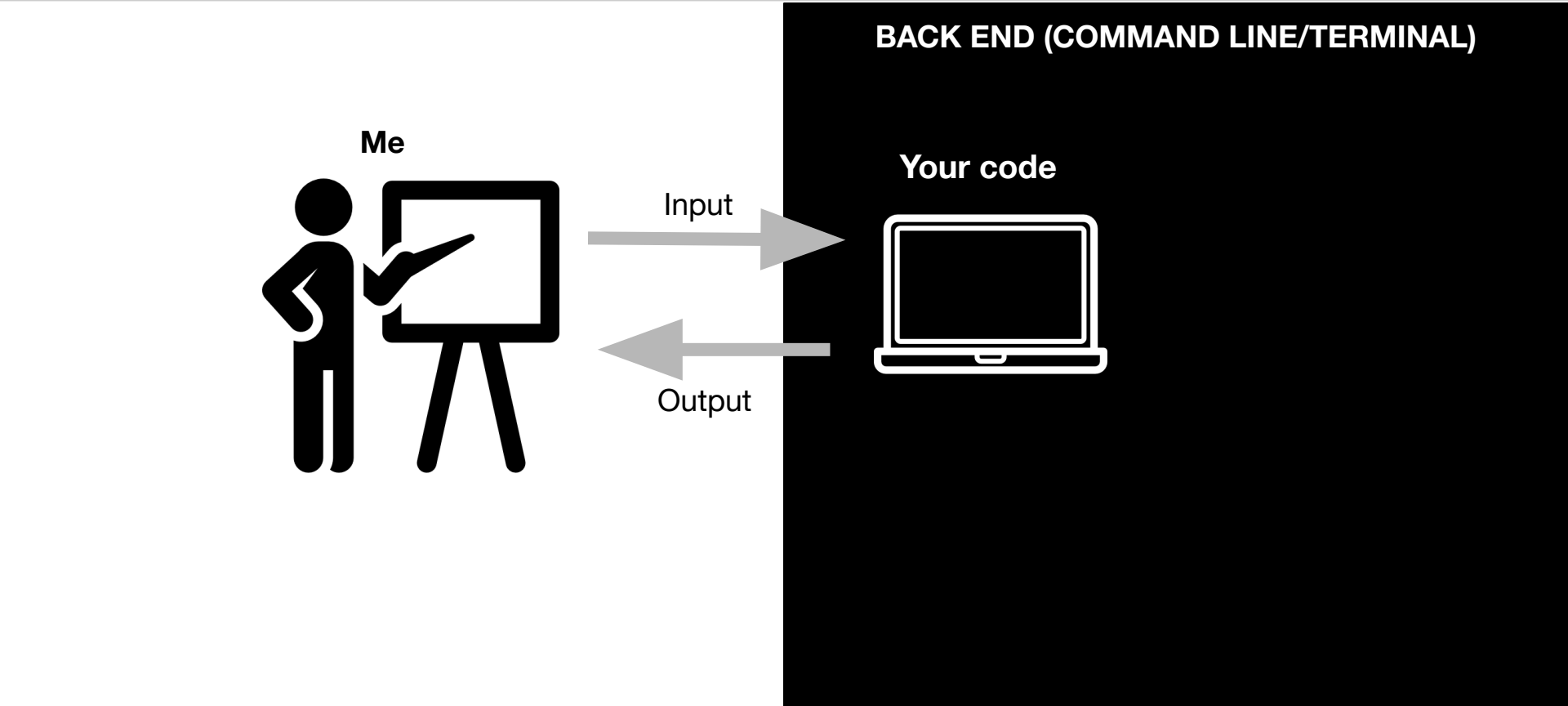
Anatomy of an Application

FRONT END (GRAPHICAL USER INTERFACE)

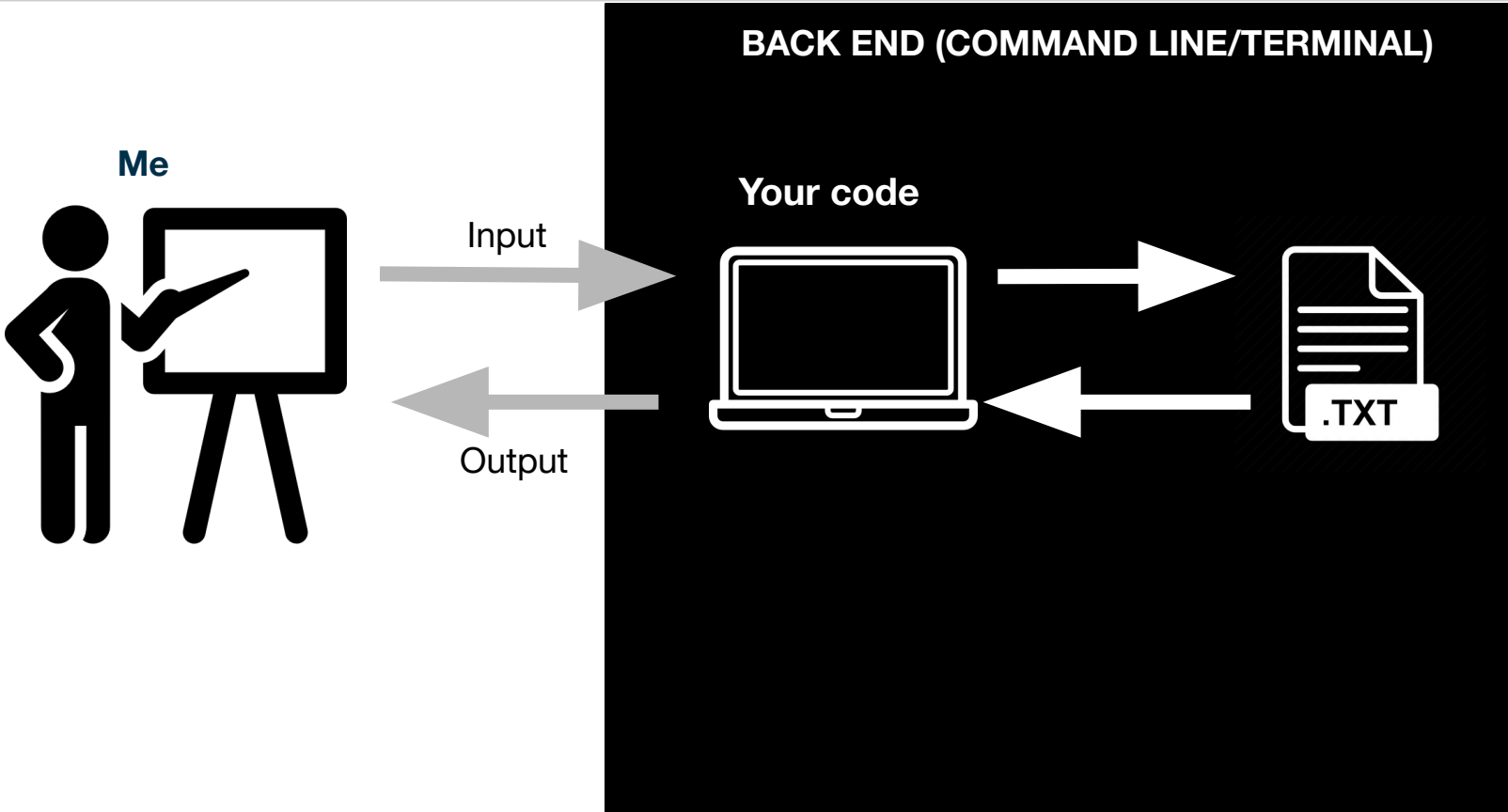
BACK END (COMMAND LINE/TERMINAL)



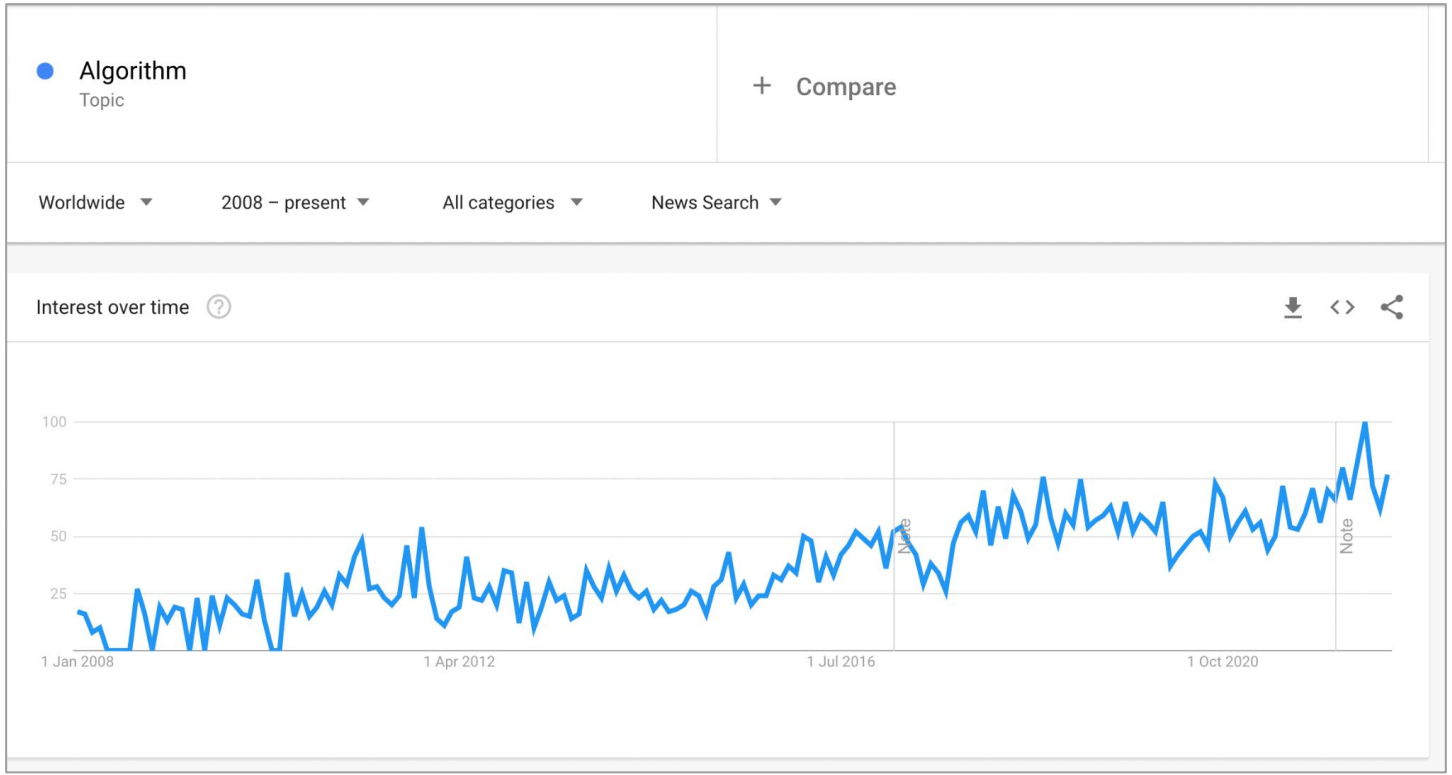
In this course, initially



In this course, later

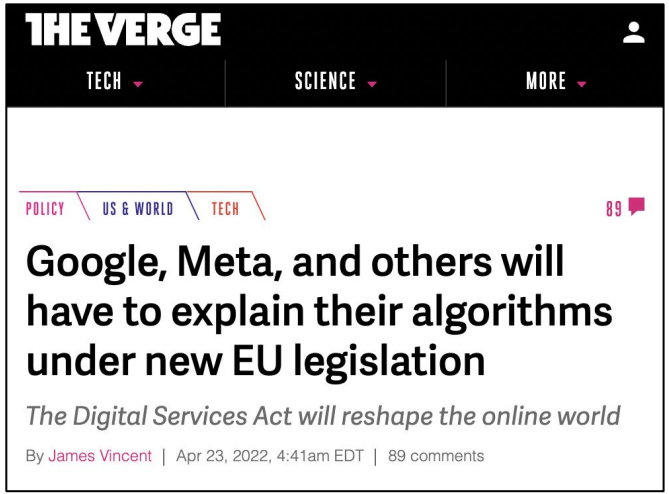


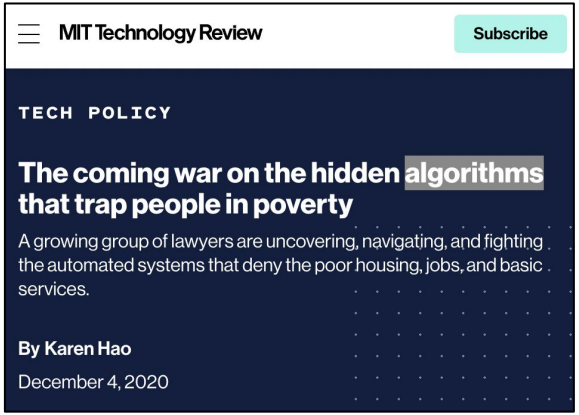
"Algorithms", in the news



https://trends.google.com/trends/explore?date=all_2008&gprop=news&q=%2Fm%2F0jpv

"Algorithms", in the news

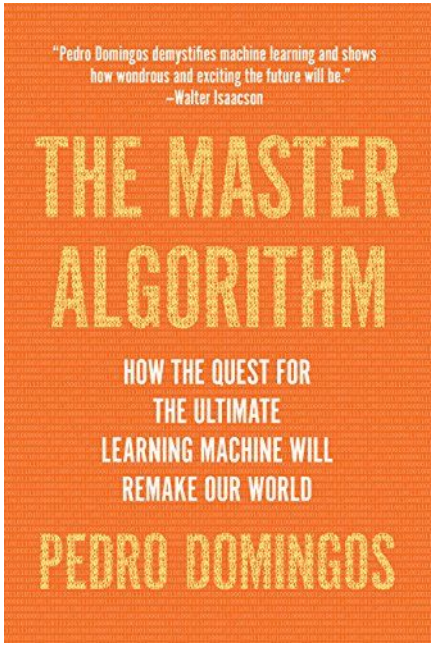


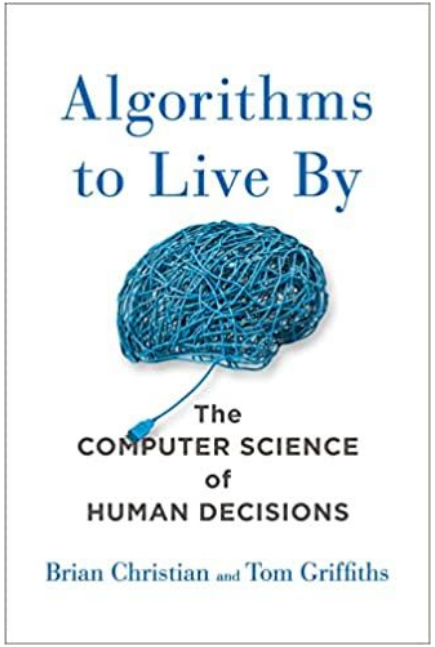


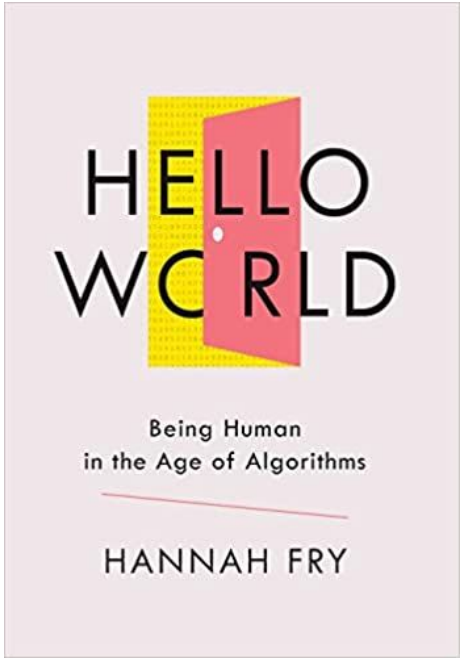
One of the best *blogs* on the internet:

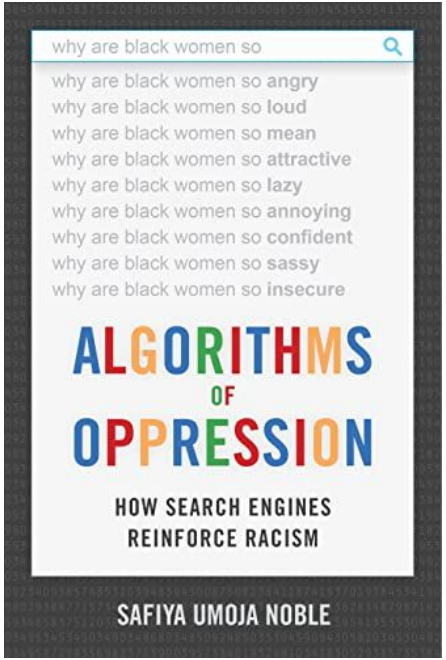
https://algopop.tumblr.com/

"Algorithms", in the bookstores



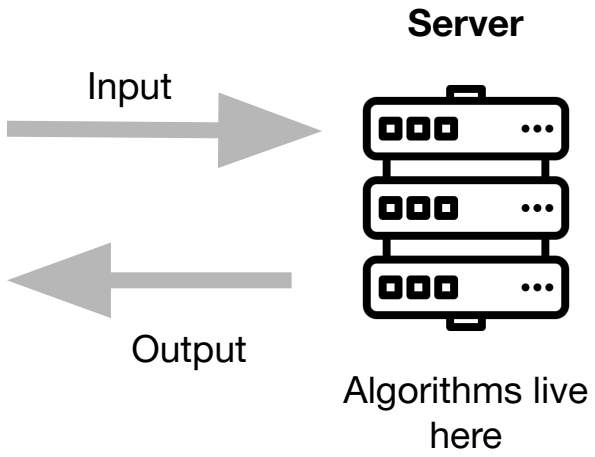






What is an "Algorithm"?

- A sequence of steps that, when performed in order, accomplishes a goal.
- In other words, a sequence of operations that process input(s) to produce output(s).
- In this course, you will implement basic algorithms



What is an "Algorithm"?

Ingredients

- 2 cups cooked chicken , chopped or shredded
- 1 can refried beans
- 1/2 cup salsa , your favorite kind
- 1 teaspoon cumin
- 1/2 teaspoon dried oregano leaves , crushed
- 1 teaspoon chili powder
- 1 cup shredded cheese , cheddar or Mexican blend
- 2 green onions , chopped
- 3 Tablespoons oil (vegetable or canola oil)
- 6 large flour tortillas

For topping:

- Salsa, sour cream and guacamole , optional

Input(s)

Algorithm / Recipe / Instructions:

1. Cook chicken breasts in frying pan until tender and no longer pink. Allow to rest for a few minutes before chopping.
2. Add refried beans, chicken, cheese, salsa, spices, and green onions to a mix bowl and mix to combine.
3. Place about 1/2 cup of the chicken mixture in the center of each tortilla.
4. Fold opposite sides over filling and roll up like a burrito.
5. **For baked chimichangas**, preheat oven to 400 degrees F. Brush chimichangas lightly with oil and bake for about 25 minutes, until golden and crispy.
6. **For pan fried chimichangas**, heat a skillet over medium heat. Once hot, add oil to skillet and place chimichangas seam side down. Turn lightly every 2-30 seconds until lightly golden on all sides.
7. Serve warm, topped with salsa and sour cream and a side of Authentic Mexican Rice .

Database / Kitchen:

- Pantry

- Appliances

- Culinary Tools

Output

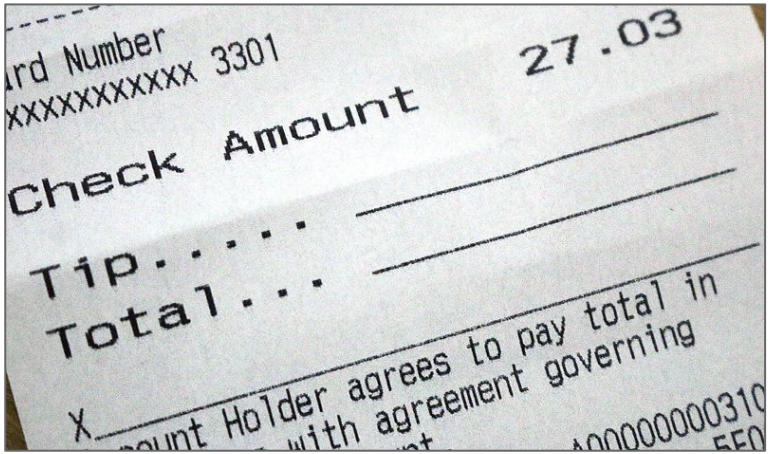
What is an Algorithm?

- That might be more of a CSC-223 (Advanced Data Structures & Algorithms) algorithm

- Let's say you just go out to a good Mexican restaurant and order a Chimichanga there

- It costs **$27.03**

- How much do you **tip?**
    - Assuming great service
    - **Without using a calculator**



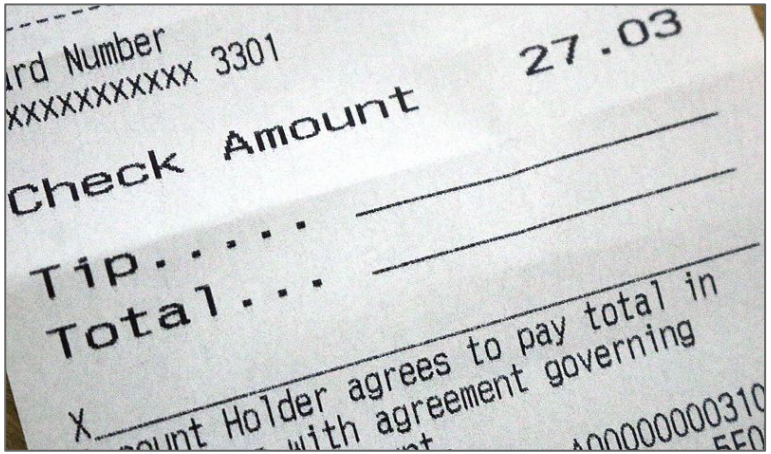
The Jane Doe Algorithm, for calculating gratuity

Input: Check Amount

Our first algorithm:

1. Move the decimal point one space, to the left
2. Round the number, from step 1
3. Double the number, from step 2

Output: ~20% of the Check Amount



The Jane Doe Algorithm, for calculating gratuity

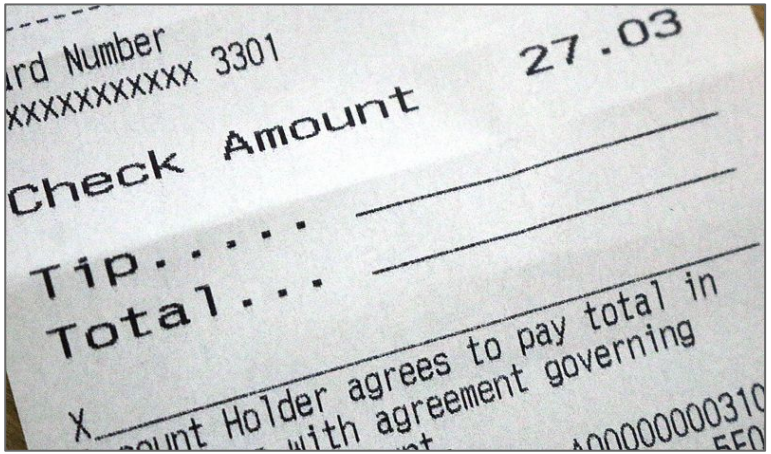
Input: Check Amount

Our first algorithm:

1. Move the decimal point to the left
   `27.03 becomes 2.703`
2. Round the number, from step 1
   `2.703 becomes 3`
3. Double the number, from step 2
   `3 becomes 6`

Output: ~20% of the Check Amount

`(6/27.03)*100 = 22.19%`



Reminder:
An algorithm is a sequence of operations that process input(s) to produce output(s).

For next time...

- In the next class, we'll see how to **implement our first algorithm in Python**
- **Please bring your laptops** to the next lab and to class
- If you haven't already, go over the following on the course **Moodle**:
    1. Course **Syllabus**
    2. Fill out the **pre-course survey**