WHAT HAVE WE (YOU!) ACCOMPLISHED IN 6.0001?

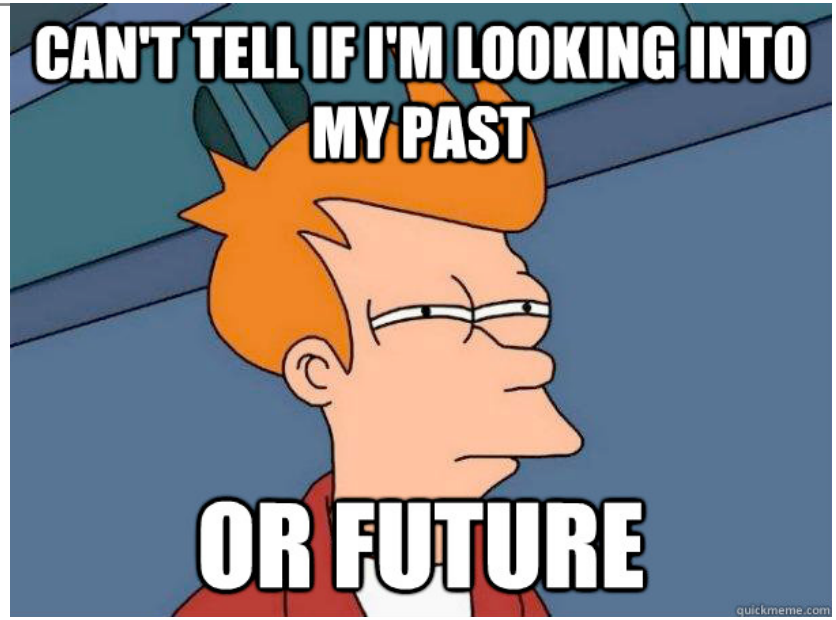
(download slides file from Stellar to follow along!)

6.0001 LECTURE 11

A “mom and apple pie” lecture

WRAPPING IT ALL UP

- where have you been?
  - what are the key topics covered in this course?
  - what are the key lessons to take from this course?
- where are you headed?
  - how might you use the knowledge you have gained?
  - what are next steps in enhancing your knowledge of computation?

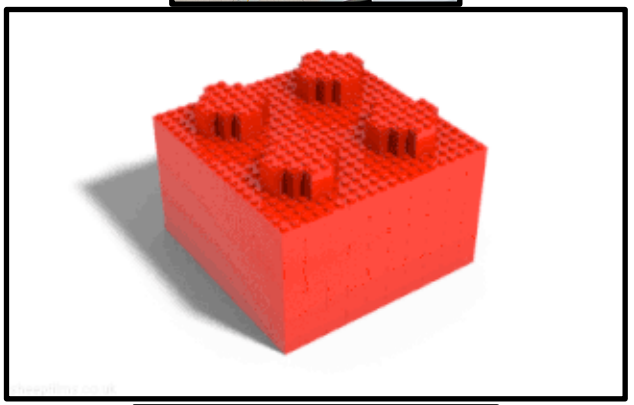


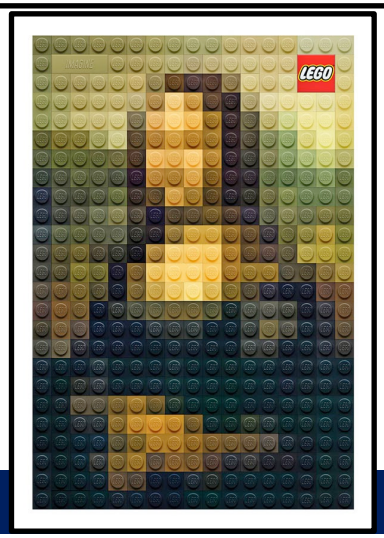
TOPICS COVERED

- represent knowledge with **data structures**
- **iteration and recursion** as computational metaphors
- **abstraction** of procedures and data types
- **organize and modularize** systems using object classes and methods
- different classes of **algorithms**, e.g., approximation, searching, sorting
- **complexity** of algorithms

TOPICS COVERED

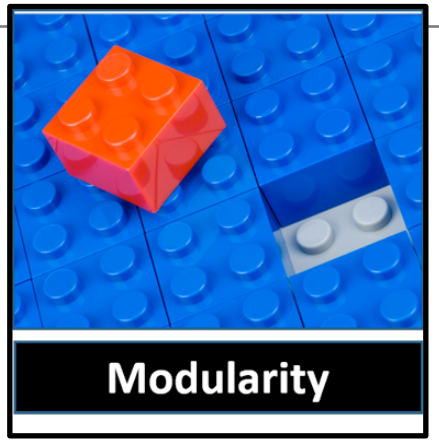
- represent knowledge with **data structures**
- **iteration and recursion** as computational metaphors
- **abstraction** of procedures and data types





TOPICS COVERED

- **organize and modularize** systems using object classes and methods



- different classes of **algorithms**, e.g., approximation, searching, sorting

- **complexity** of algorithms

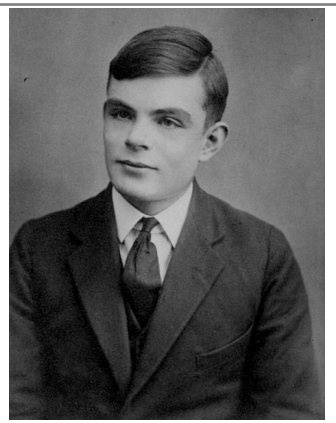
HIGH LEVEL VIEW OF COURSE

- learn computational modes of thinking
- master the art of computational problem solving
- make computers do what you want them to do

Hope we have started you down the path to being able to think and act like a computer scientist

HOW DO COMPUTER SCIENTISTS THINK?

- they think computationally
  - abstractions, algorithms, automated execution
- just like the three r's: reading, 'riting, and 'rithmetic – computational thinking is becoming a fundamental skill that every well-educated person will need



Alan Turing

Ada Lovelace

Grace Hopper

Don Knuth

I ♥ 6.0001

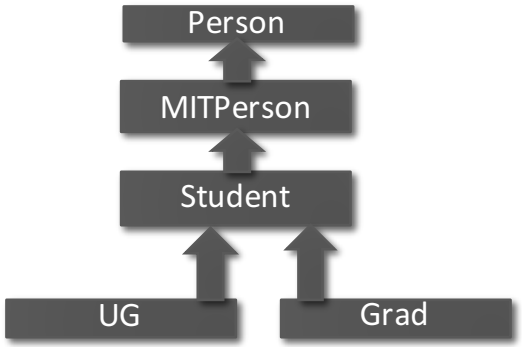
COMPUTATIONAL THINKING: THE PROCESS

- identify or invent useful abstractions
  - suppress details, create interfaces between modules
- formulate solution to a problem as a computational experiment using abstractions
- design and construct a sufficiently efficient implementation of experiment
- validate experimental setup (i.e., debug it)
- run experiment
- evaluate results of experiment
- repeat as needed

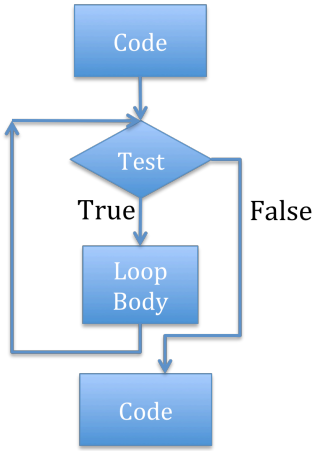
Will see these ideas in greater detail in 6.0002, if you are staying for that part of the class

THE THREE A'S OF COMPUTATIONAL THINKING

- Abstraction
  - choosing the right abstractions
  - operating in multiple layers of abstraction simultaneously
  - defining relationships between the abstraction layers



- Automation
  - think in terms of mechanizing our abstractions
  - mechanization is possible – because we have precise and exacting notations and models; and because there is some "machine" that can interpret our notations



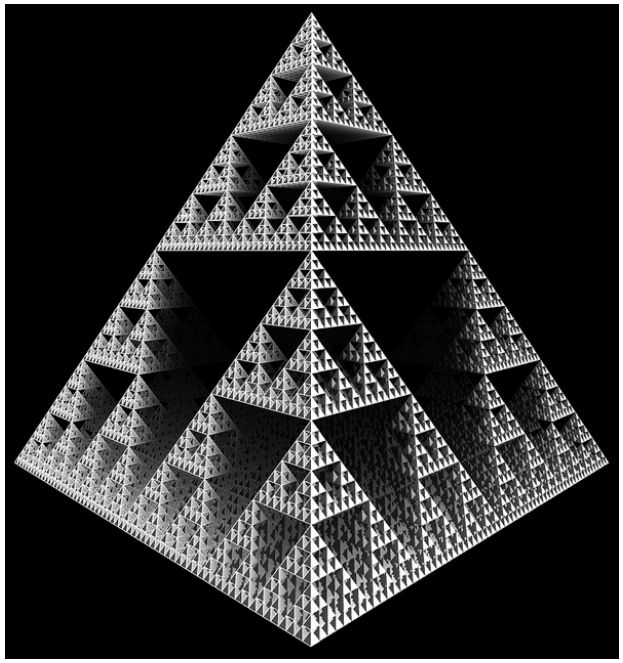
- Algorithms
  - language for describing automated processes
  - also allows abstraction of details
  - language for communicating ideas & processes

```
def mergeSort(L, compare = operator.lt):
    if len(L) < 2:
        return L[:]
    else:
        middle = int(len(L)/2)
        left = mergeSort(L[:middle], compare)
        right = mergeSort(L[middle:], compare)
        return merge(left, right, compare)
```

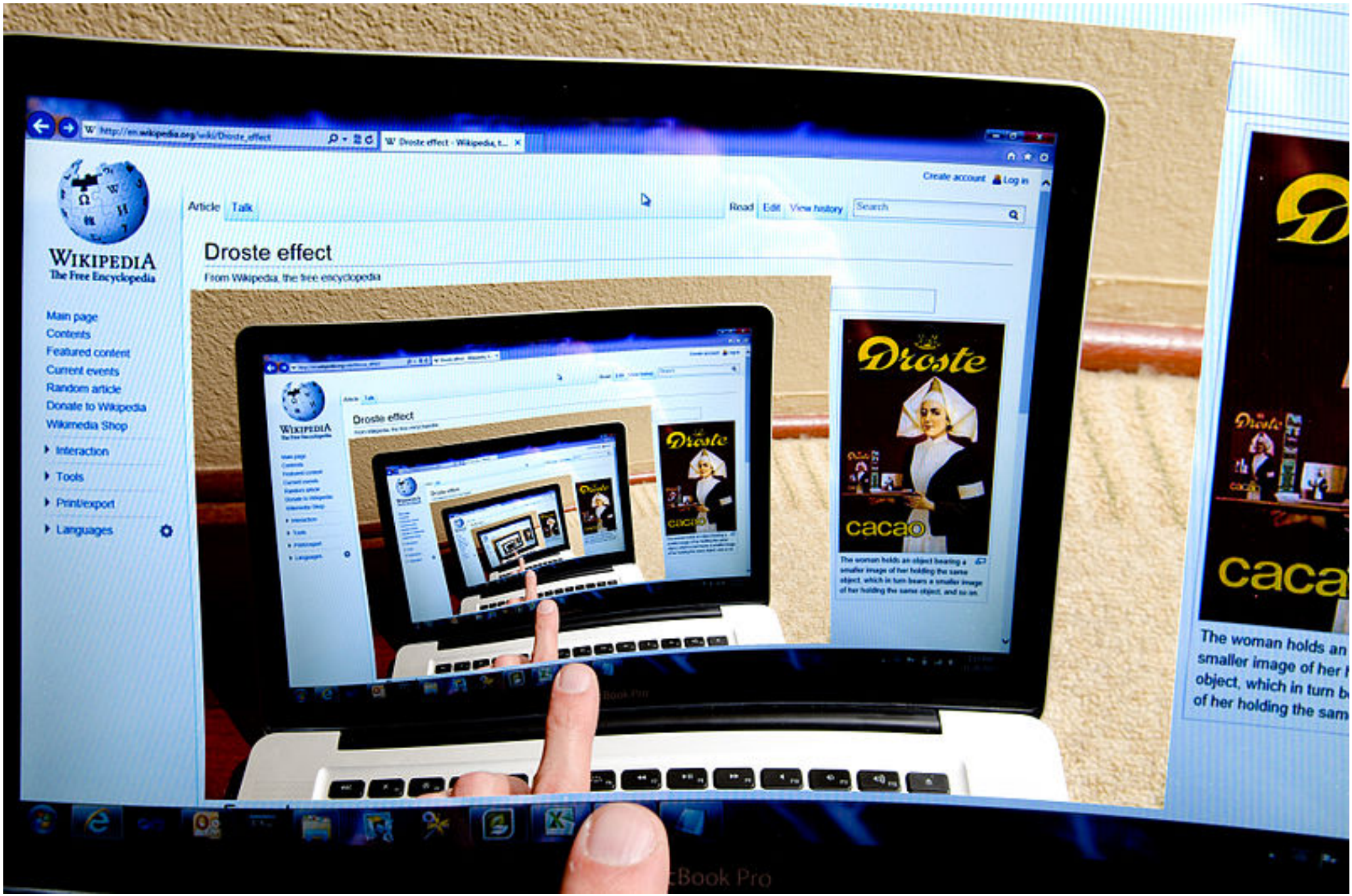
SOME KEY ASPECTS OF COMPUTATIONAL THINKING

- how difficult is this problem and how best can I solve it?
  - theoretical computer science gives precise meaning to these and related questions and their answers
- thinking recursively
  - reformulating a seemingly difficult problem into one which we know how to solve
  - reduction, embedding, transformation, simulation

O(log n) ; O(n) ; O(n log n) ; $O(n^2)$; $O(c^n)$

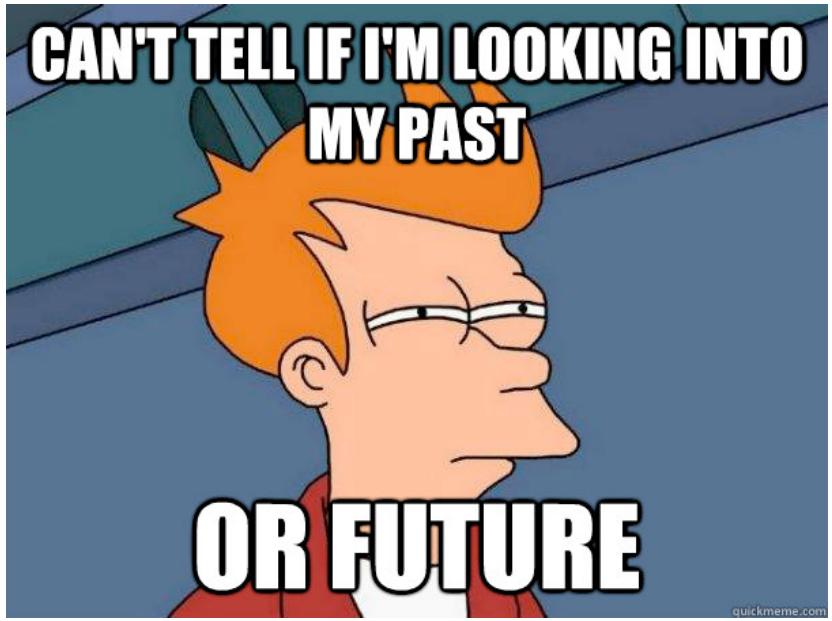


CC-BY Image Courtesy of Robson#


Droste effect
From Wikipedia, the free encyclopedia
WIKIPEDIA
The Free Encyclopedia
Droste
cacao

WRAPPING IT ALL UP

- where have you been?
  - what are the key topics covered in this course?
  - what are the key lessons to take from this course?
- where are you headed?
  - how might you use the knowledge you have gained?
  - what are next steps in enhancing your knowledge of computation?



NEXT STEPS

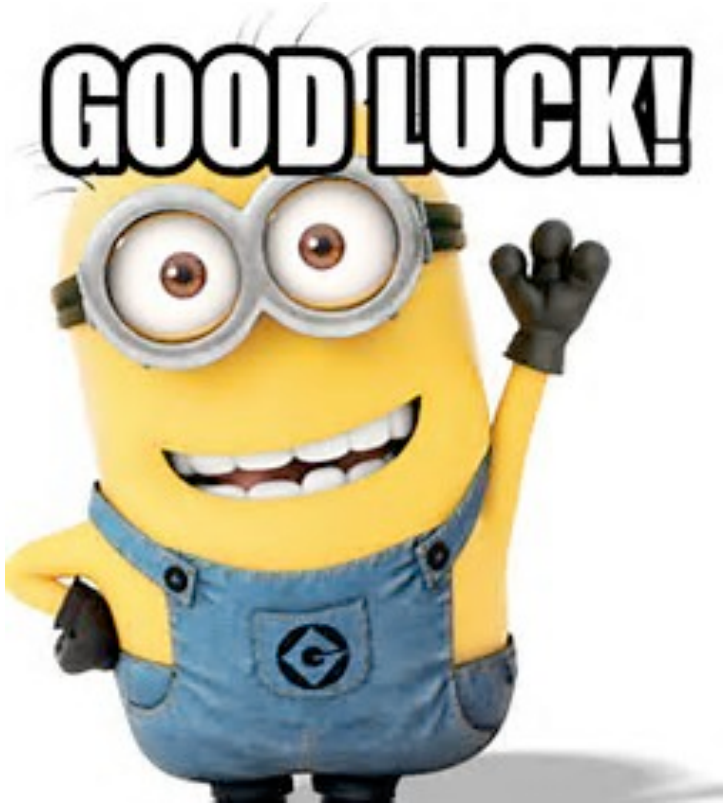
- look for ways to apply what you have learned:
  - can you use algorithmic ideas in your professional life?
    - how might abstraction, or computational experiments, be used to improve what you hope to do once you start or join a company?
    - how can these ideas help you pursue your choice of discipline more effectively?
  - can you use algorithmic approaches in your personal life?
    - organizing your personal finance records, your family historical records, your student group activities

GOOD LUCK!

- however you choose to use computational thinking, we hope that it becomes a useful tool for you:
  - as a way of approaching professional problems
    - e.g., running computational experiment to simulate physical or biological or financial or other problems
  - as a basis for understanding the impact of computation in everyday life
    - e.g., what is the power of machine learning methods in solving complex problems
  - as a language for communicating ideas
    - e.g., explaining ideas as concise steps in an algorithmic process, independent of whether one actually implements it

By OsamaK (Own work) [Public domain], via Wikimedia Commons

And “Good Luck” on the Exam!



On to quiz review