



COMP 150: Basics of Computer Organization

Week 04, Lecture 07



Agenda

1. Review: Loops and Conditionals
2. Review: Top-Down
3. Practice!

Conditionals and Loops

- if/elif/else
- while
- for
- try/except



Conditionals and Loops

- if/elif/else
- while
- for
- try/except

```
# IF/ELIF/ELSE
if i < 10:
    print("less than 10")
elif i < 100 and i > 10:
    print("greater than ten and less than 100")
else:
    print("greater than 100")

# WHILE
x = 0
while x < 10:
    print(x)
    x = x + 1

# FOR
l = [1,2,3,4]
for y in l:
    print(y)

# TRY/EXCEPT
a = 1
b = 0
try:
    a / b
except:
    print("division didn't work")
```

Top-Down

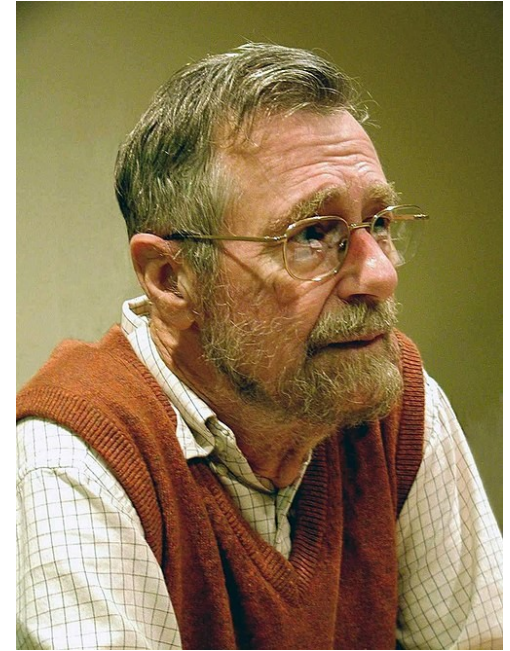
In essence:

Programs should solve problems.

We can't write decent programs if we are mired in details.

Solution?

Start generic (pseudocode!), and specify until it works.



Edsger W. Dijkstra

Father of computer
programming

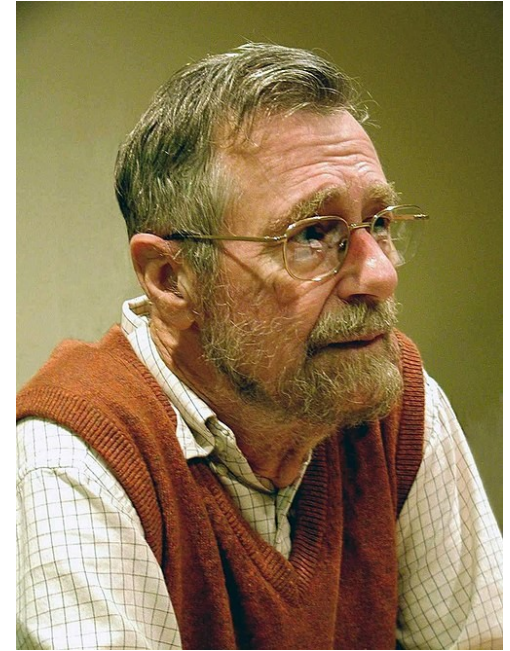
Top-Down

Start with *pseudocode*:

Use plain English to describe what the code should do, ex:
`show the first 10 prime numbers`

Then turn it into more *syntactic code*:

Take some of the plain English and turn it into code, ex:
`print(the first 10 prime numbers)`



Edsger W. Dijkstra

Father of computer
programming

Top-Down

And keep going until it's something the program can do:

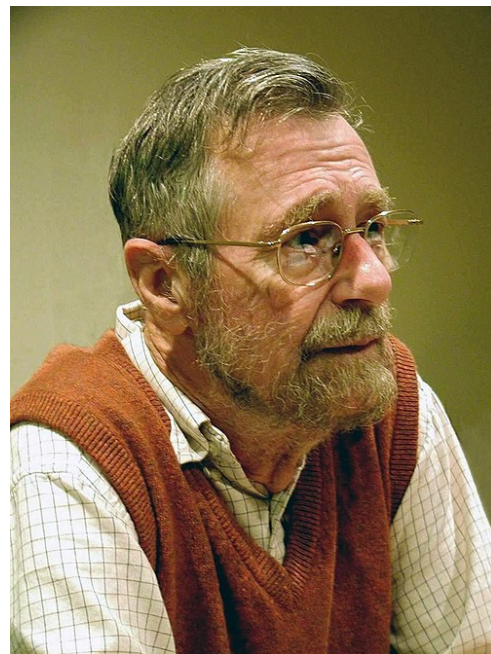
```
ten_primes = the first ten prime numbers
```

```
print(ten_primes)
```

```
ten_primes = [2,3,5,7,11,13,17,19,23,29]
```

```
print(ten_primes)
```

Done!



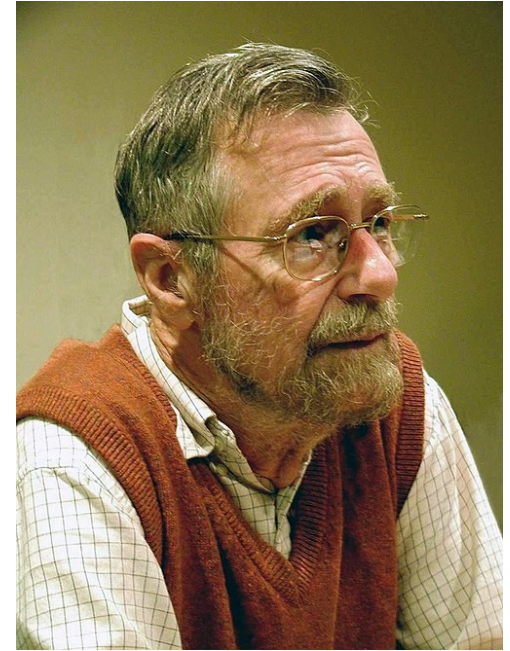
Edsger W. Dijkstra

Father of computer
programming

Top-Down

1. Solve the problem - feel free to be generic!
2. Pick some part, and make it more specific
 - a. Smaller steps?
 - b. Store something in a variable?
 - c. Repetition?
 - d. Call a function?
3. Stuck? Search for one of your steps
4. Do this until the problem is accurately solved by the program

When you're done, if it's a mess, or if it's too slow, you now have a new problem to solve!



Edsger W. Dijkstra

Father of computer
programming

Practice!

Fizz Buzz

Fizz Buzz

Canonical problem - similar to Hello World

- Give the program an integer n
- For every number $i < n$,
 - If i is divisible by 3, output “Fizz”
 - If i is divisible by 5, output “Buzz”
 - If i is divisible by both 3 and 5, output “FizzBuzz”
 - If i is not divisible by 3 nor by 5, print the value of i

Example:

Input: $n = 3$

Output: [1 2 Fizz]

Input: $n = 10$

Output: [1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz]

Input: $n = 20$

Output: [1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 Fizz 19 Buzz]

image from <https://www.geeksforgeeks.org/python/fizz-buzz-in-python/>

Alternation

Canonical problem - similar to Hello World

- Give the program a list
- Print every other element in the list, starting by printing the first

image from <https://www.geeksforgeeks.org/python/fizz-buzz-in-python/>