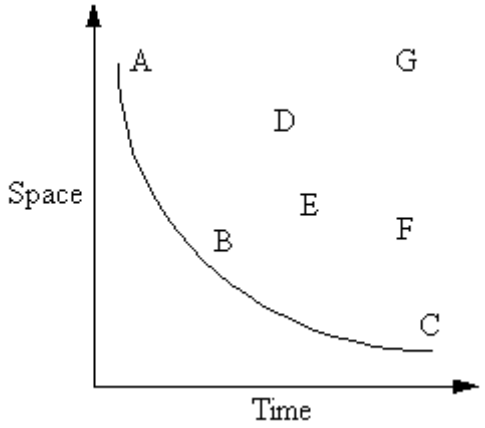
Big Picture: Computer Science

- Computer Science is no more about computers than astronomy is about telescopes. (Edsger W. Dijkstra)
- Computation: To determine what can be computed and how it can best be computed
- The term “computer science” is a misnomer
- Computed?: the manipulation of data by a computational procedure (algorithm)
- Best?: algorithm that makes the most efficient use of our computational resources
- Computational resource?
  processor & memory



Big Picture 2: Computers and Programs

- **Computer: a machine manipulating symbols**
- **Symbols: any discrete entity: letters, numbers…**
- **Programs: rules telling how to manipulate symbols**
  - Develop algorithm
  - Write computer program
  - Convert (compile) to binary (the language of computer)
  - Load to memory (cache) (done by OS)
  - Run/execute: program takes over the computer (done by OS)

Examples of Computers

Raspberry Pi:
starting at $10

IBM summit:
~3Million processors
10MW power
https://www.top500.org/

D-Wave (Vancouver)
2000Qbit quantum computer

IBM-Q

https://en.wikipedia.org/wiki/List_of_quantum_processors

Big Picture: Programming

Computer Programming is a field that involves the methodology behind the programming, software abstraction, algorithms, data structures, design, testing, and maintenance of computer software. (Wikiversity)

Program: tells computer what to do

Catches:

1. expressing things in symbols

2. Comp's don't speak English

3. Comp languages can only refer

to ojb's in their world, not the natural world

4. Comp's can do simple manipulations, hence detailed instructions necessary

On the bright side:

1. World of comp's is a simple one: built on six key concepts

2. Prog languages contain very few words (33 keywords in Python)

3. A large set of techniques avail. for common situations, no need to start from scratch.

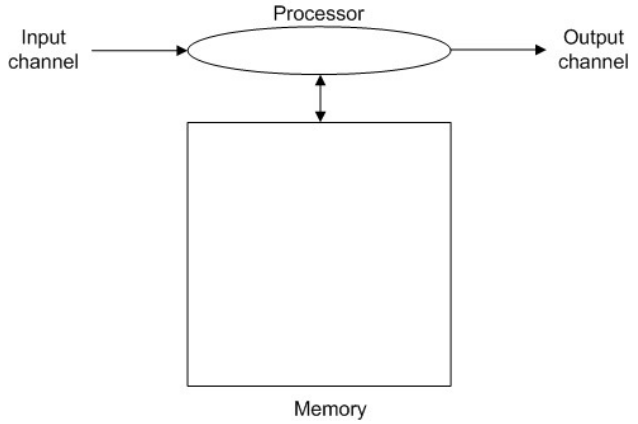
The world of a computer: Six key op’s

The six operations consist of three elementary operations:

- **input**
- **processing, and**
- **output**

and three control structures,

- **sequence**
- **selection, and**
- **repetition**



Input, processing, and output

```
# input
speed = input()
duration = input()

#processing
distance = speed * duration

#output
print distance
```

Sequential Execution

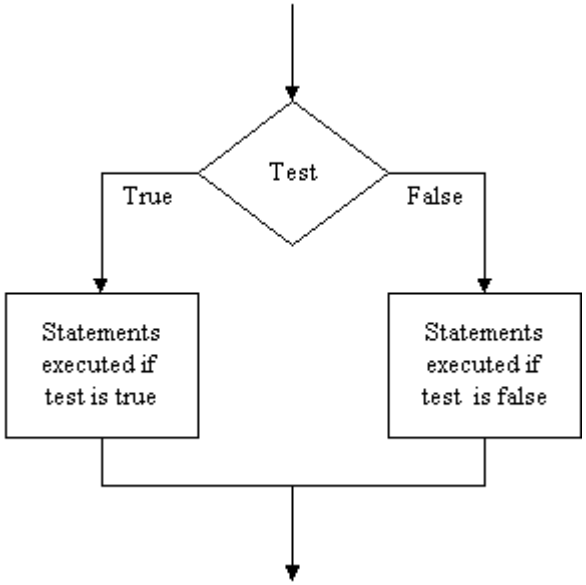
By design, a program will execute sequentially, otherwise a program won't work:

```
# This works as intended
speed = input()
duration = input()
distance = duration * speed
print distance
```

```
# This won't work
speed = input()
print distance
distance = duration * speed
duration = input()
```

Selection

```
#
if boxers_weight > 90:
    print(“Heavy weight”)
else:
    print(“Not heavy weight”)
```

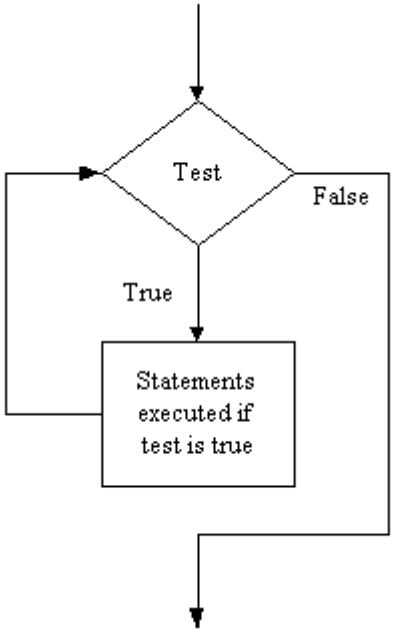


Repetition

Some procedures require the ability to repeat a group of instructions:

- process the records in a file until the last line

- approximate f(x) until successive error is less than 0.001



while error **>** 0.001**:**

    approx_new=... #newton-raphson

    error= abs(approx_old-approx_new)

Exercise: play with Python shell

First program: Fahrenheit to Celsius conversion

The problem:

We need to convert a temperature measured on the Fahrenheit scale to its equivalent on the Celsius scale

Now we will work through the solution of this problem as follows:

1) Solving the problem by hand to make sure we know how to do it.

2) Consciously identifying our solution method, i.e. the solution algorithm.

3) Translating that algorithm into Python.

4) Entering and running the resulting program and verifying that it works.

First program: Fahrenheit to Celsius conversion

The problem:

We need to convert a temperature measured on the Fahrenheit scale to its equivalent on the Celsius scale

Step1: pick a temperature in Fahrenheit , and convert to Celsius by hand.

Using Wikipedia: C = (F – 32)x5/9

Pick F=50 degrees -> C=(50-32)*5/9=10 degrees Celsius

First program: Fahrenheit to Celsius conversion

The problem:

We need to convert a temperature measured on the Fahrenheit scale to its equivalent on the Celsius scale

Step1: pick a temperature in Fahrenheit , and convert to Celsius by hand.

Using Wikipedia: C = (F – 32)x5/9

Pick F=50 degrees -> C=(50-32)*5/9=10 degrees Celsius

Step2: write it down as simple steps (i.e. algorithm) and think in terms of input, processing, output steps

1. get temp in F from user and save to a variable
2. calc temp in C
3. output C

First program: Fahrenheit to Celsius conversion

The problem:

We need to convert a temperature measured on the Fahrenheit scale to its equivalent on the Celsius scale

Step3: Translate the algorithm into Python

Python 2.x

```
temp_in_f = input()
temp_in_c = (temp_in_f - 32) * 5 / 9
print temp_in_c
```

Python 3.x

```
temp_in_f = eval(input())
temp_in_c = (temp_in_f - 32)*5/9
print(temp_in_c)
```

Step4: Run and test for a few F values

First program: Fahrenheit to Celsius conversion

Can this code be improved ?

```
temp_in_f = eval(input())
temp_in_c = (temp_in_f - 32)*5/9
print(temp_in_c)
```

First program: Fahrenheit to Celsius conversion

Can this code be improved ?

- Naming variables
- User interaction
- Documentation

```
temp_in_f = eval(input())
temp_in_c = (temp_in_f - 32)*5/9
print(temp_in_c)
```

Naming variables

We did a decent job in picking variable names:

- **descriptive variable names (which enhance program readability)**
- **yet short, for convenience,**
- **all lowercase to adhere to Python convention**

What not to do (not descriptive!)

```
x = input()
y = (x - 32) * 5 / 9
print y
```

User interface and user interaction

What is wrong here?

```
$ python3 f2c.py
40
4.444444444444445
```

User interface and user interaction

What is wrong here?

```
$ python3 f2c.py
40
4.444444444444445
```

Would user realize they are supposed to enter a value?

Would user think 4.44.. is output or think its just a random number or something?

User interface and user interaction

We need to provide instructions to the user, and explain output.

```
1
2 print("This program converts temperatures from Fahrenheit to Celsius.")
3 print("Enter a temperature in Fahrenheit (e.g. 10) and press Enter.")
4 temp_in_f = eval(input("Temperature in Fahrenheit: "))
5 #https://en.wikipedia.org/wiki/Fahrenheit
6 temp_in_c = (temp_in_f - 32)*5/9
7 print(temp_in_f, "degrees Fahrenheit =", temp_in_c, "degrees Celsius.")
8
```

```
$ python3 f2c_v2.py
This program converts temperatures from Fahrenheit to Celsius.
Enter a temperature in Fahrenheit (e.g. 10) and press Enter.
Temperature in Fahrenheit: 40
40 degrees Fahrenheit = 4.444444444444445 degrees Celsius.
```

User interface and user interaction

```
1
2 print("This program converts temperatures from Fahrenheit to Celsius.")
3
4 print("Enter a temperature in Fahrenheit (e.g. 10) and press Enter.")
5
6 temp_in_f = eval(input("Temperature in Fahrenheit: "))
7
8 #https://en.wikipedia.org/wiki/Fahrenheit
9 temp_in_c = (temp_in_f - 32)*5/9
10
11 print(temp_in_f, "degrees Fahrenheit =", temp_in_c, "degrees Celsius.")
12
```

A few things to note here:

- variables & messages (string literals) can be combined in "print" statement using comma

- "input" statement can include a string literal as prompt

- we added blank lines to improve readability

Documentation

Necessary step for producing complete program

- **Meta comments: who, when, why created the code**
  - Use '#' (also possible to use docstring)
- **Inline comments: anything that puzzled you and you might not remember in 6 month from now**
- **Charts & user manual (for large projects)**

Documentation: Example

```
1
2 # f2c.py: converts a given temperature in Fahrenheit to Celsius
3 # CPSC 128 Example program
4 # S. Bulut, Spring 2018-19
5
6 print("This program converts temperatures from Fahrenheit to Celsius.")
7
8 print("Enter a temperature in Fahrenheit (e.g. 10) and press Enter.")
9
10 temp_in_f = eval(input("Temperature in Fahrenheit: "))
11
12 #https://en.wikipedia.org/wiki/Fahrenheit
13 temp_in_c = (temp_in_f - 32)*5/9
14
15 print(temp_in_f, "degrees Fahrenheit =", temp_in_c, "degrees Celsius.")
16
```

Second program: c2f

Now its your turn to write a program that converts from Celsius to Fahrenheit

- follow the previously mentioned best practices

- save your code under your git repo

- once you are done do

- git commit -m ‘my first code’ c2f.py
- git push