



Technical University of Lodz
Institute of Electronics

Algorithms and Data Structures

Introduction for Biomedical Engineering IFE Students

Łódź 2013





General Information

Lecturers



Andrzej Materka

andrzej.materka[at]p.lodz.pl

<http://amaterka.pl>

Phone: 42 631 26 26

Room: 322, building B9

Consultations: Monday 2:15pm



Marek Kociński

marek.kocinski[at]p.lodz.pl

<http://www.eletel.p.lodz.pl/kocinski/>

Phone: 42 631 26 17

Room: 218, building B9

Consultations: check the callendar

https://poczta2011.p.lodz.pl/service/user/marek.kocinski@p.lodz.pl/studenci_2012.html



General Information

- **Lecture/Tutorial:** 30h (15 weeks x 2 hours)
- **Venue:** room 413, building B9, Wólczańska 211/215
- **Credits:** 1 ECTS point (1 ECTS point = 25 - 30 h)
- **Self work:** 25 - 30 hours
- **Programming Language:** Python
- **Tools:** Enthought Canopy environment



Most popular languages (according to GitHub)

GitHub is a [web-based hosting service](#) for software development projects that use the [Git revision control](#) system. GitHub offers both paid plans for private repositories, and free accounts for open source projects. As of May 2011, GitHub was the most popular open source code repository site.^[3]

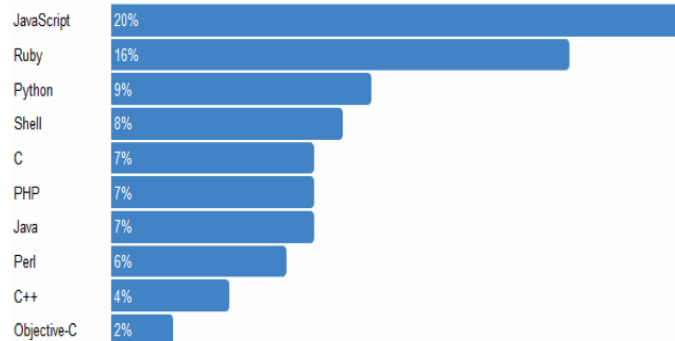
GitHub Inc. was founded in 2008 and is based in San Francisco, California.^[4]

In July 2012, the company received US\$100 Million in [Series A](#) funding, primarily from [Andreessen Horowitz](#).^{[5][6][7]}

Contents [\[hide\]](#)

- [1 Description](#)
- [2 Statistics](#)
- [3 Limitations and Constraints](#)
- [4 See also](#)
- [5 Notes](#)
- [6 References](#)
- [7 External links](#)

GitHub Top Languages 2011



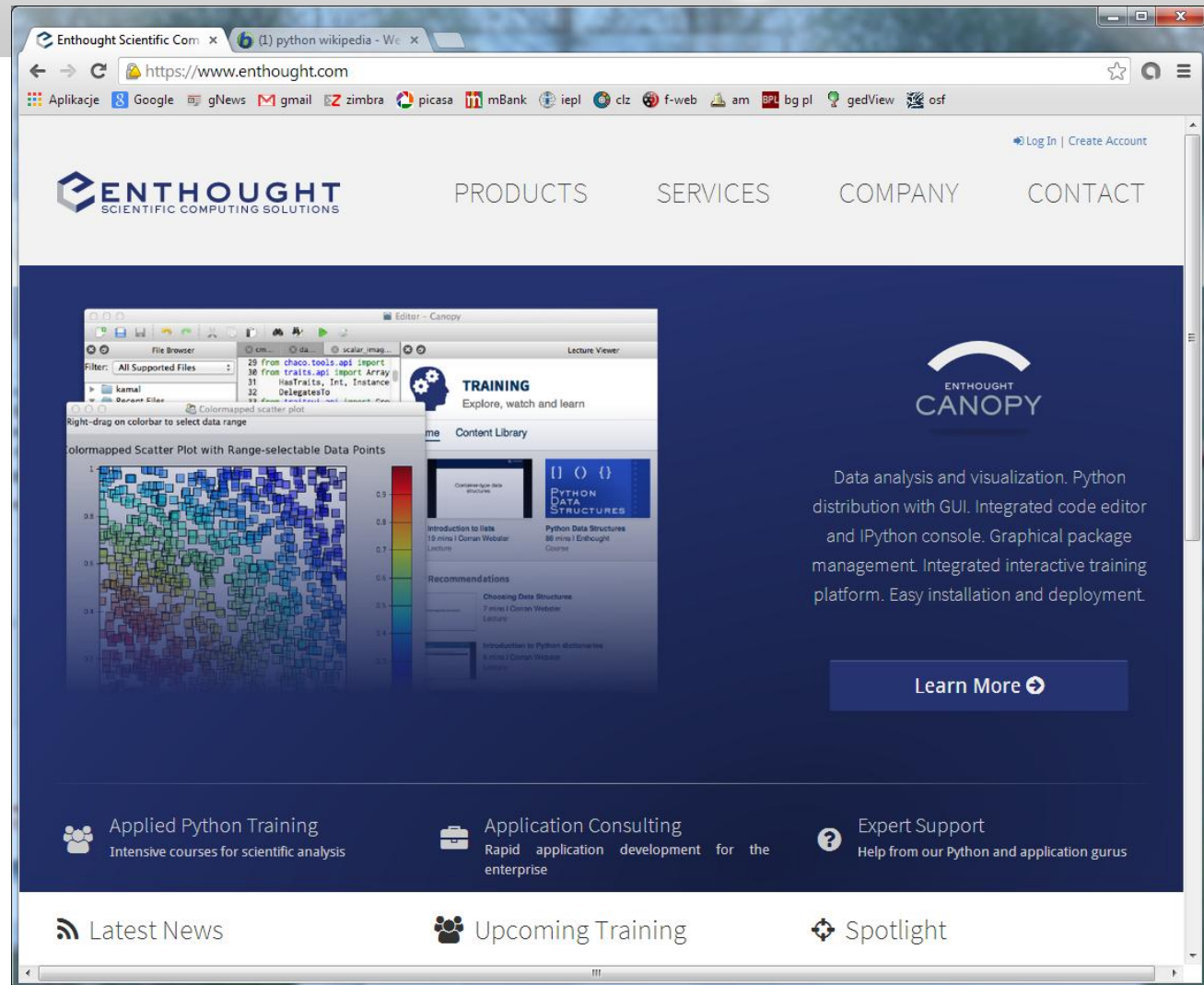
GitHub

github
SOCIAL CODING

URL	GitHub.com [link]
Slogan	Social Coding (for all)
Commercial?	Yes
Type of site	collaborative revision control
Registration	Required
Available language(s)	English
Owner	GitHub, Inc. [link]
Launched	April 2008 ^[1]
Alexa rank	▼ 317 (September 2012) ^[2]

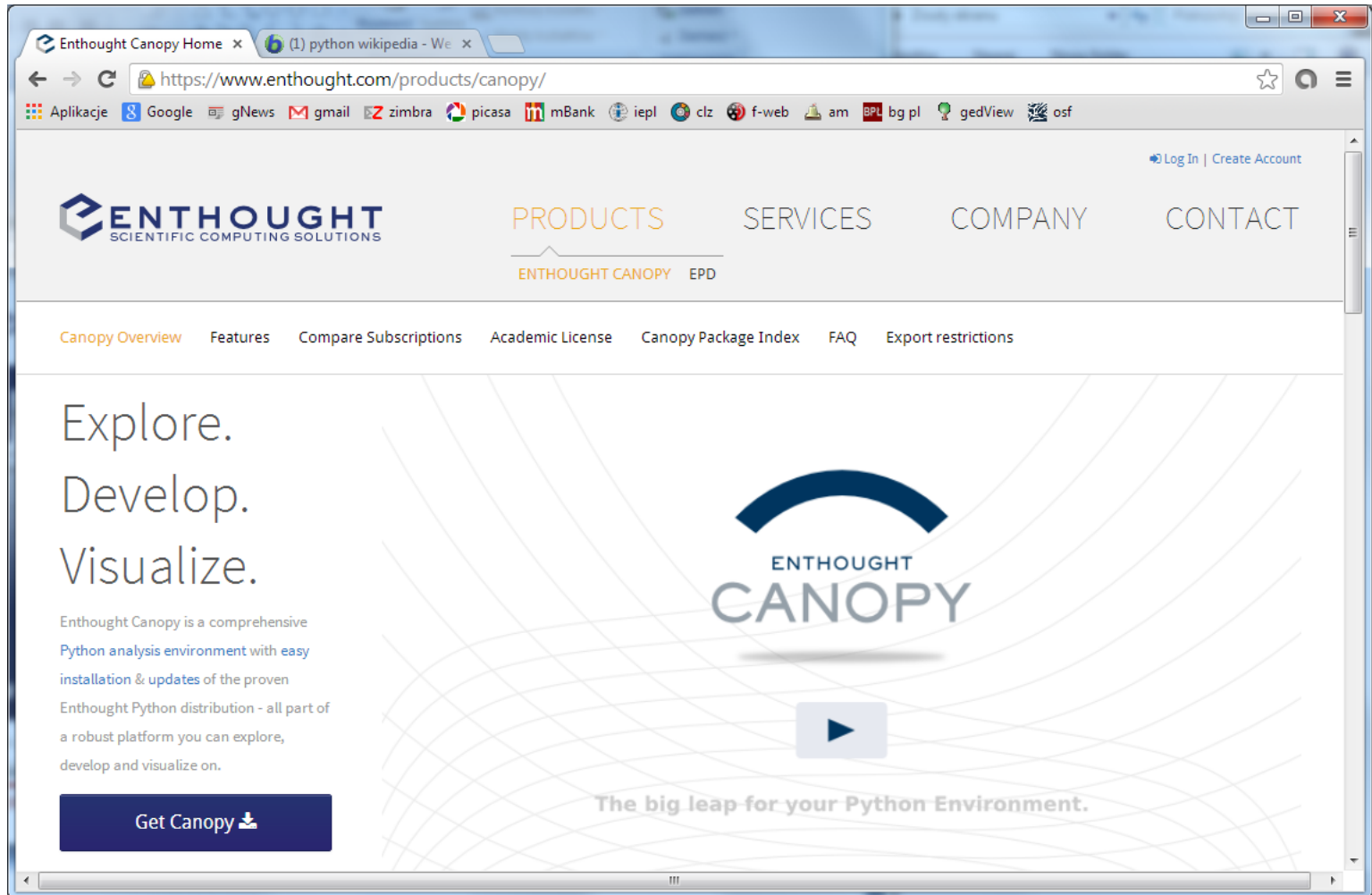


www.enthought.com





www.enthought.com/products/canopy/

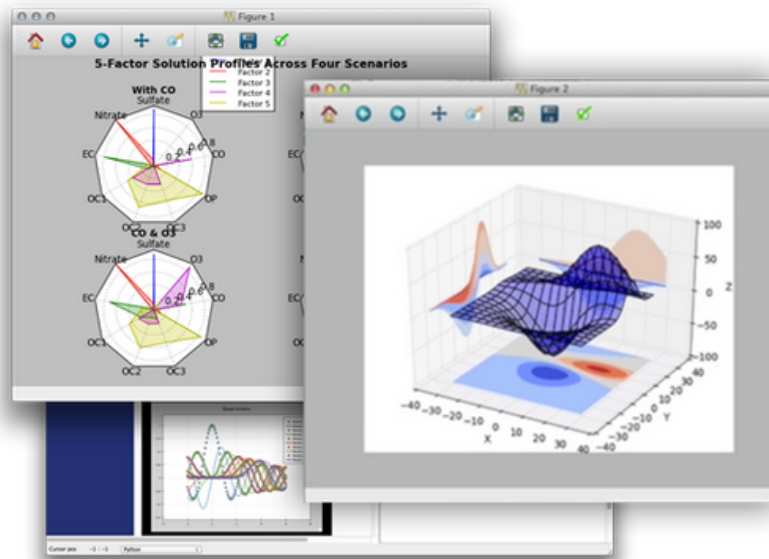




www.enthought.com/products/canopy/

[Canopy Overview](#) [Features](#) [Compare Subscriptions](#) [Academic License](#) [Canopy Package Index](#) [FAQ](#) [Export restrictions](#)

What's in it for you?



Scientists and Engineers

A comprehensive, Python-based analysis desktop & Python distribution, Canopy provides an open and intuitive environment for scientific and analytic computing. Since it's Python, your algorithms, scripts and programs will never be locked into a proprietary language. And with the analysis desktop, data analysis, scripting and plotting are more straightforward.

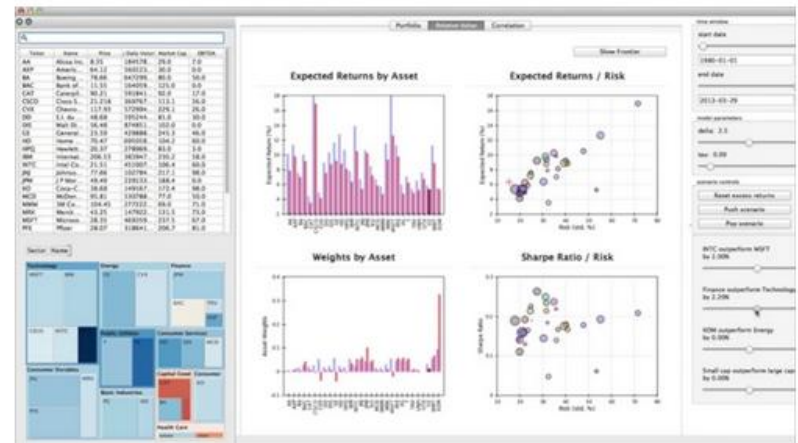


www.enthought.com/products/canopy/

Canopy Overview Features Compare Subscriptions Academic License Canopy Package Index FAQ Export restrictions

Quantitative and Data Analysts

Python spans from algorithm development and testing on the desktop to teams developing web-server applications. With the Canopy desktop and Python distribution, data ingestion, manipulation and analysis are simplified, and algorithm prototyping and testing are streamlined.





www.enthought.com/products/canopy/

[Canopy Overview](#) [Features](#) [Compare Subscriptions](#) [Academic License](#) [Canopy Package Index](#) [FAQ](#) [Export restrictions](#)



Portable Power

NORTHROP GRUMMAN



Enterprise

As a comprehensive Python-based analysis environment, Canopy puts powerful, yet very cost-effective, tools in the hands of analysts, scientists and engineers. As a robust application platform, it streamlines technical computing application development and deployment for your organization and for your customers.

With the popular, intuitive Python language and the comprehensive Canopy application platform, your organization can deploy new applications, algorithms and analysis tools much faster than with standard software languages and platforms. Users, especially "power" users, can extend and innovate with scripting and open platform APIs, driving the creation and sharing of innovative techniques and tools.



Who is using Python?

python™

NASA

python™

CENTRAL INTELLIGENCE AGENCY
UNITED STATES OF AMERICA

Jan Koprowski <jan.koprowski@gmail.com> Politechnika Gdańska, FTIMS – Informatyka Stosowana 5



Who is using Python?

Kto używa języka Python



NOKIA
CONNECTING PEOPLE





Who is using Python?

Kto używa języka Python



YAHOO! GROUPS

YAHOO! MAPS
INDIA BETA

YAHOO! OUR CITY
INDIA BETA



Walt Disney
**FEATURE
ANIMATION**

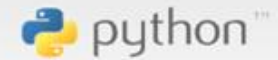


© PhotographyBLOG

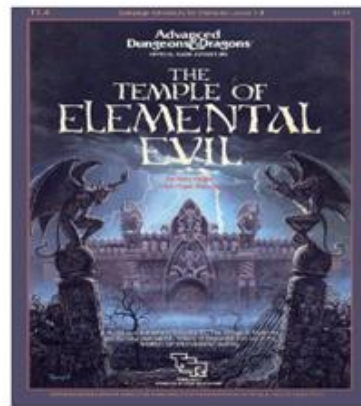
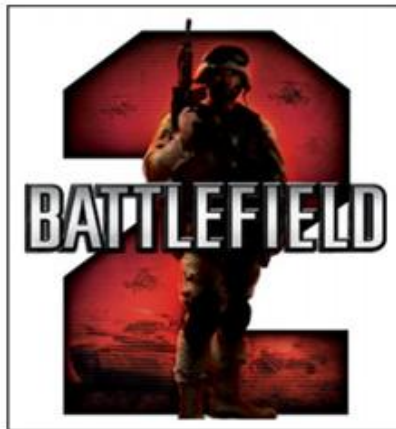


Who is using Python?

Kto używa języka Python



SID MEIER'S
CIVILIZATION IV



Jan Koprowski <jan.koprowski@gmail.com> Politechnika Gdańska, FTIMS – Informatyka Stosowana

8



Python and other programming languages

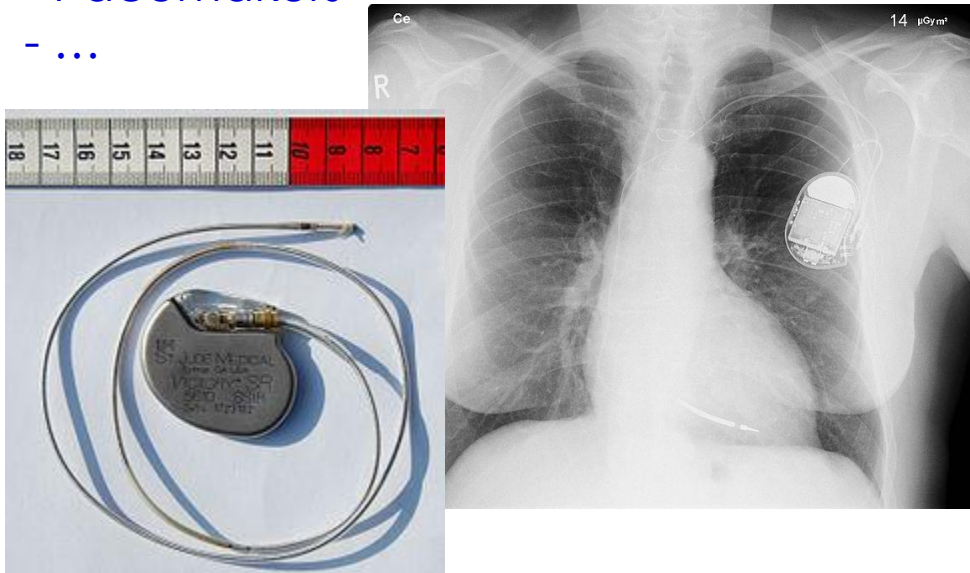
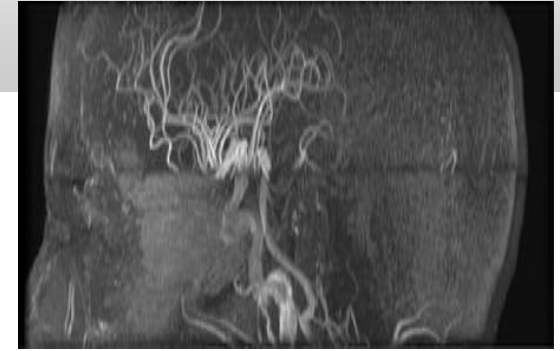
Python vs World





Computers in Medicine

- HealthCare IT Systems
- Medical Imaging (CT, MRI,...)
- Image Analysis/Segmentation
- Medical Signal Analysis
- Surgery Navigation Aids
- Pacemakers
- ...



http://en.wikipedia.org/wiki/Artificial_cardiac_pacemaker



Computers in Medicine



Laptops, Netbooks and Tablets



Desktops and Workstations



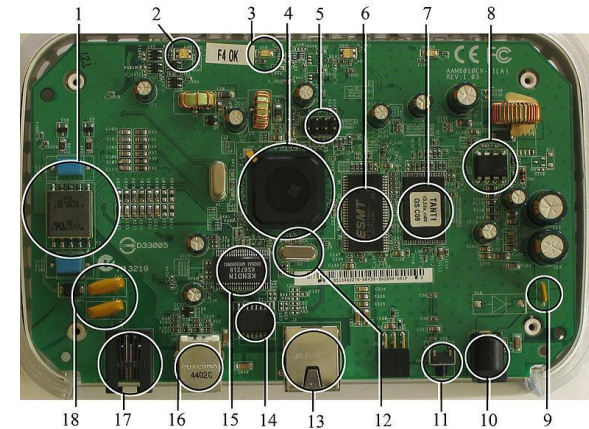
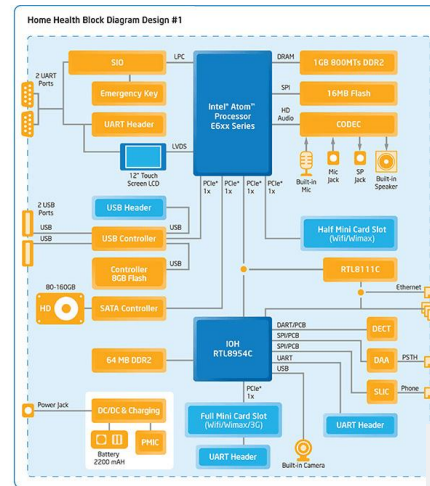
Servers, Storage and Networking



Software and Peripherals

<http://content.dell.com/us/en/healthcare/healthcare-solutions>

- HealthCare IT Systems
- Medical Imaging (CT, MRI,...)
- Image Analysis/Segmentation
- Medical Signal Analysis
- Surgery Navigation Aids
- Pacemakers
- ...



Embedded systems



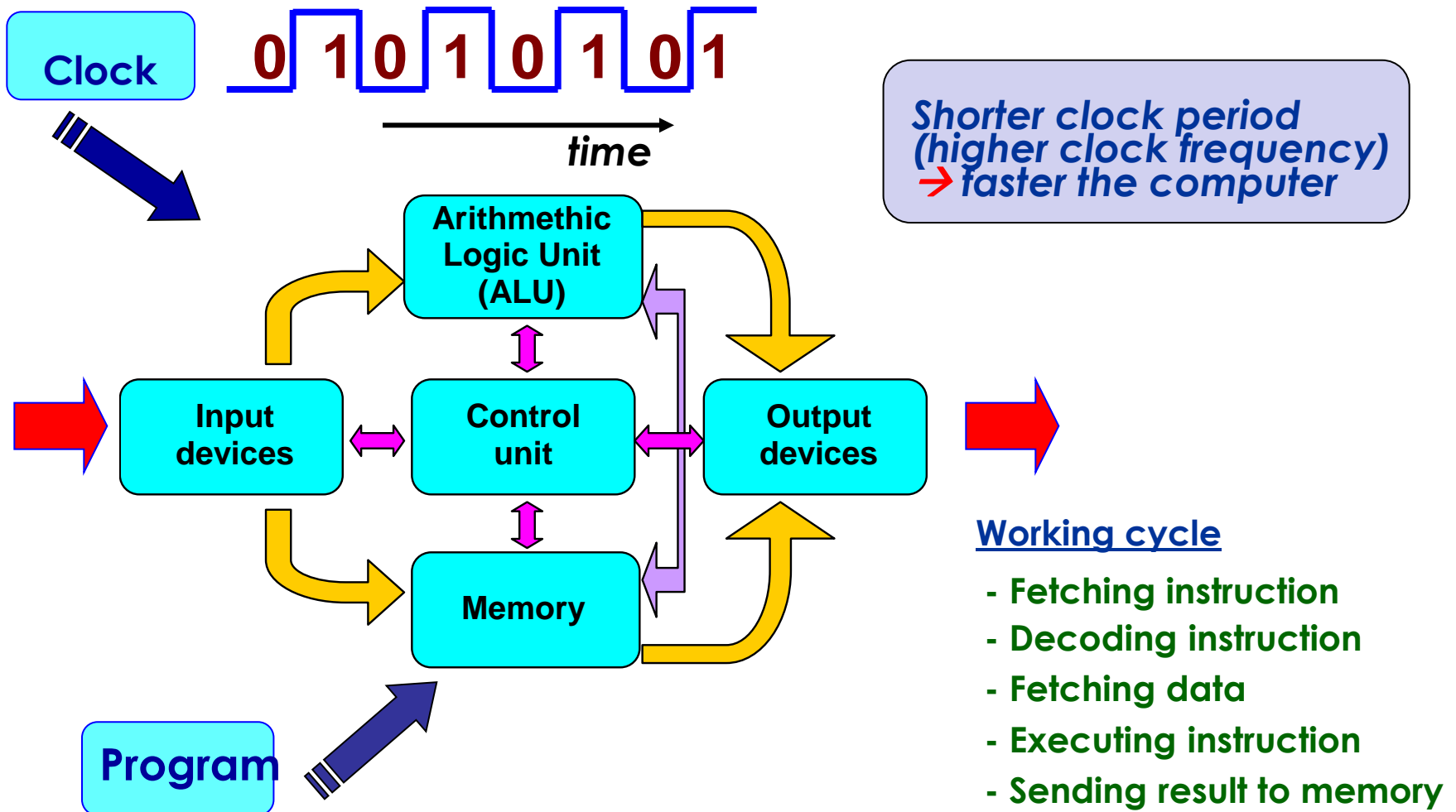
- What is a computer in those applications?
- What functions does it perform?
- Why do we use computers?





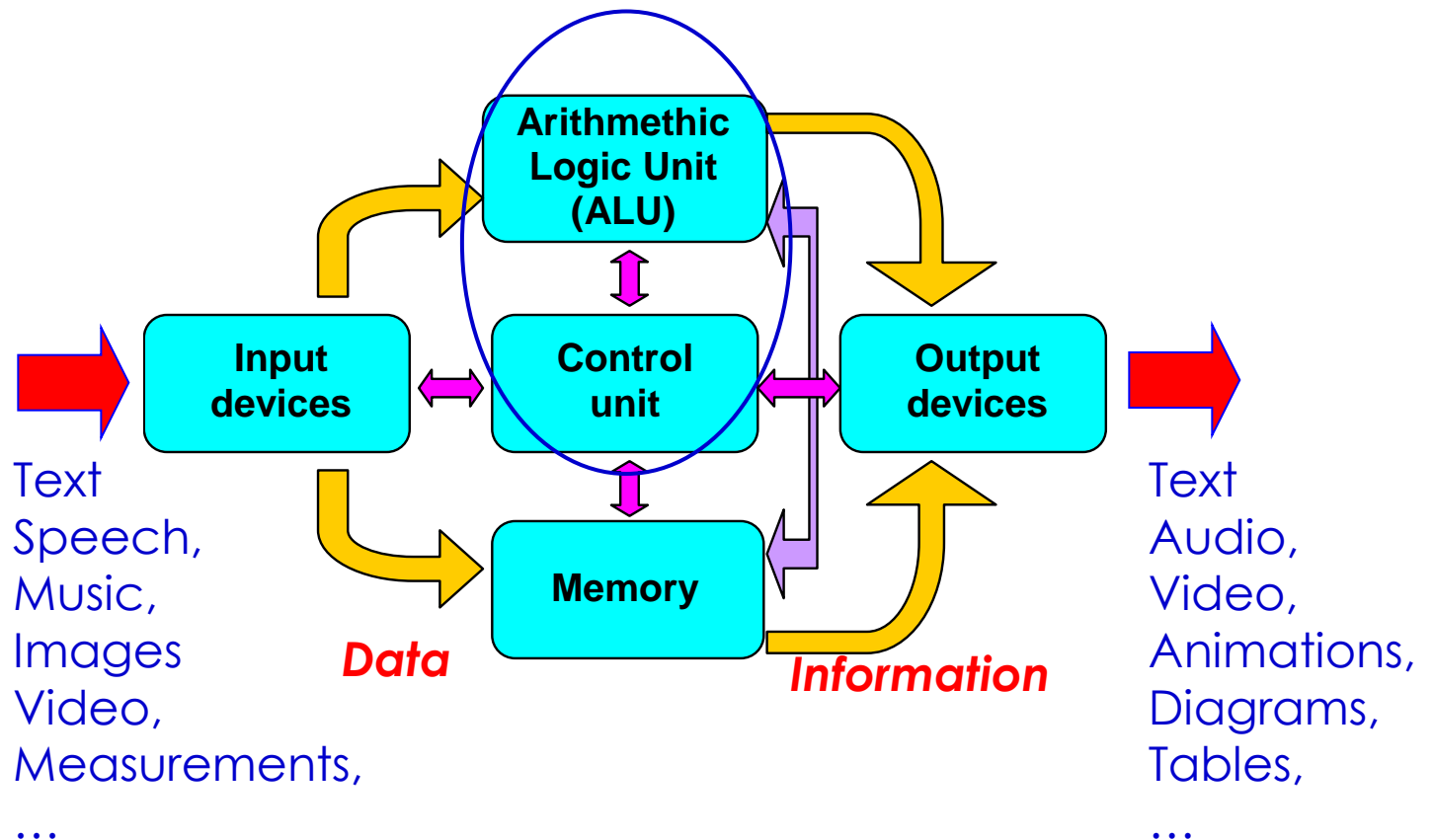
Basics of Computer Architecture

John von Neumann (1903-1957)



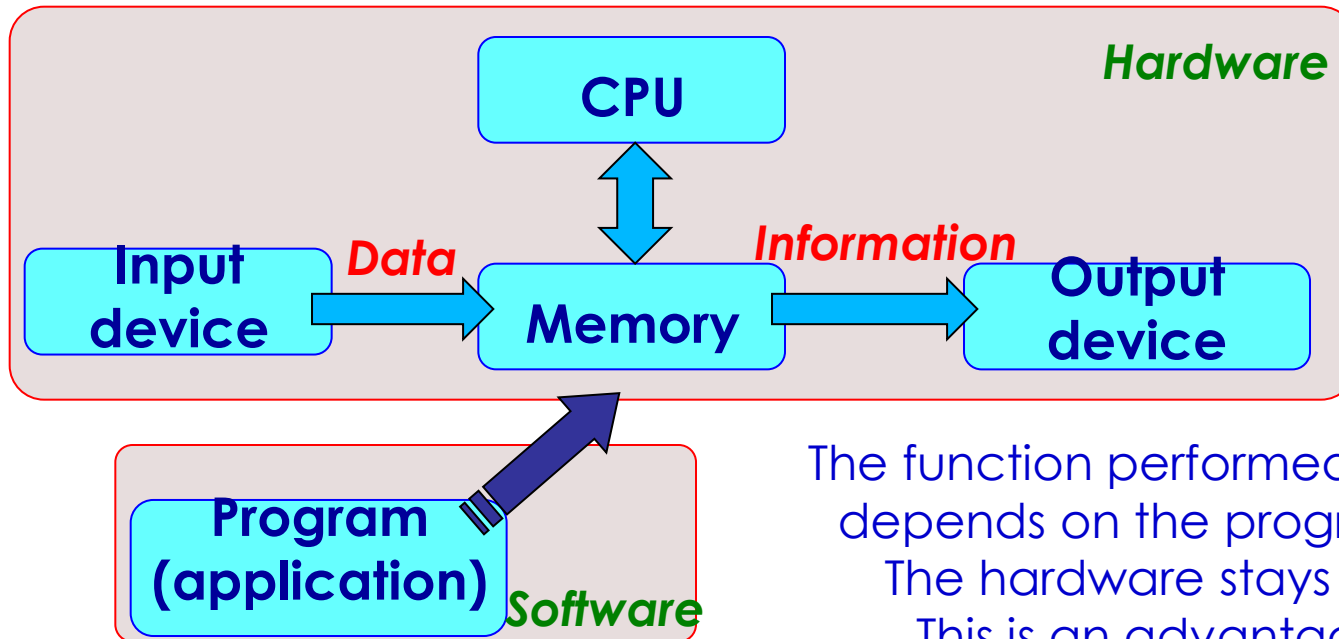
Basics of Computing

(Arithmetic Logic Unit + Control Unit) = Central Processing Unit (CPU)





The need for programming

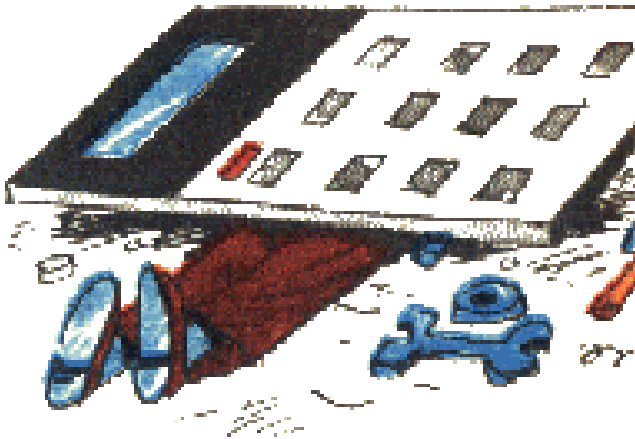


The function performed by a computer depends on the program (software).
The hardware stays all the same.
This is an advantage of digital computers (functional flexibility).



Basic assumptions

- The data are represented by binary numbers
- Time of execution of elementary instructions is very short
- The processing elements are of very small size



Representation							
decimal				binary			
10	1	8	4	2	1		
	0					0	
	1					1	
	2			1	0		
	3			1	1		
	4		1	0	0		
...			...				
1	5	1	1	1	1		

- decomposition into simple operations
- robustness to noise



Basic binary operations

Examples of binary operations

Rules of addition

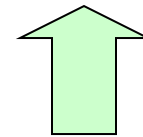
0	0	1	1
+0	+1	+0	+1
<hr/>	<hr/>	<hr/>	<hr/>
0	1	1	10

6×5:

$$\begin{array}{r} 110 \\ \times 101 \\ \hline 110 \\ 000 \\ + 110 \\ \hline 11110 \end{array}$$

Text encoding

! - 00100001
\$ - 00100100
A - 01000001
B - 01000010
Z - 01011010
a - 01100001
m - 01101110



Possibility of
operations on symbols

Computer memory



Capacitor

Plates

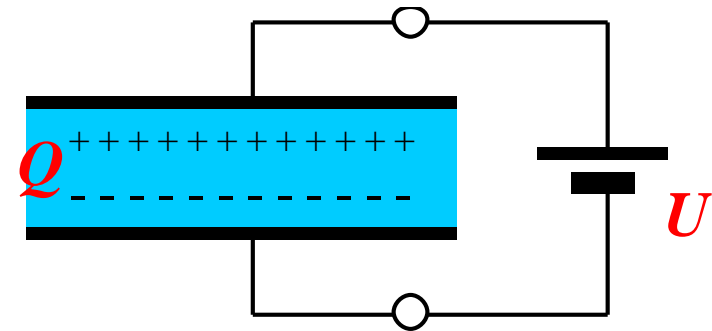
A

d

Insulator (ϵ)

Capacity of a capacitor

$$C = \frac{\epsilon A}{d}$$



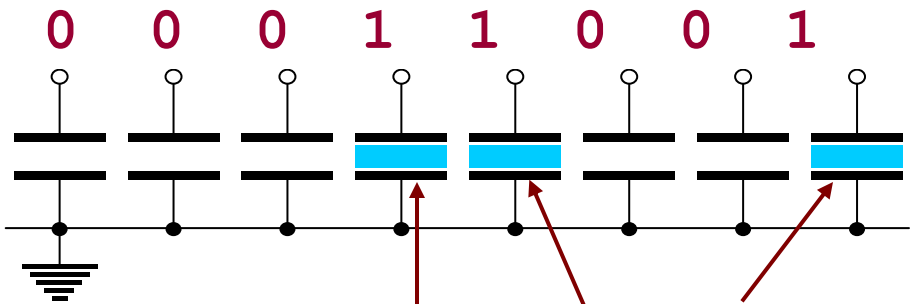
$$Q = CU$$

Charge

Capacity

Voltage

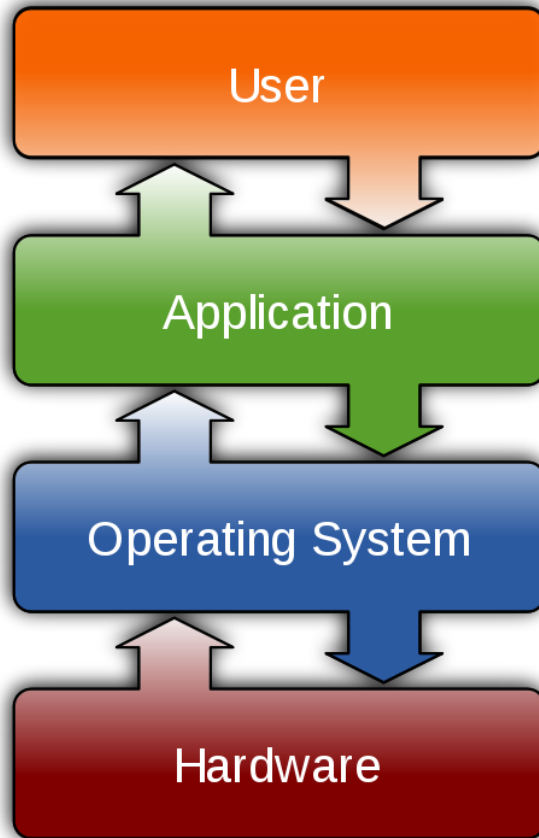
1 byte of memory (8 bits)



$$16 + 8 + 1 = 25$$



Operating system (OS)



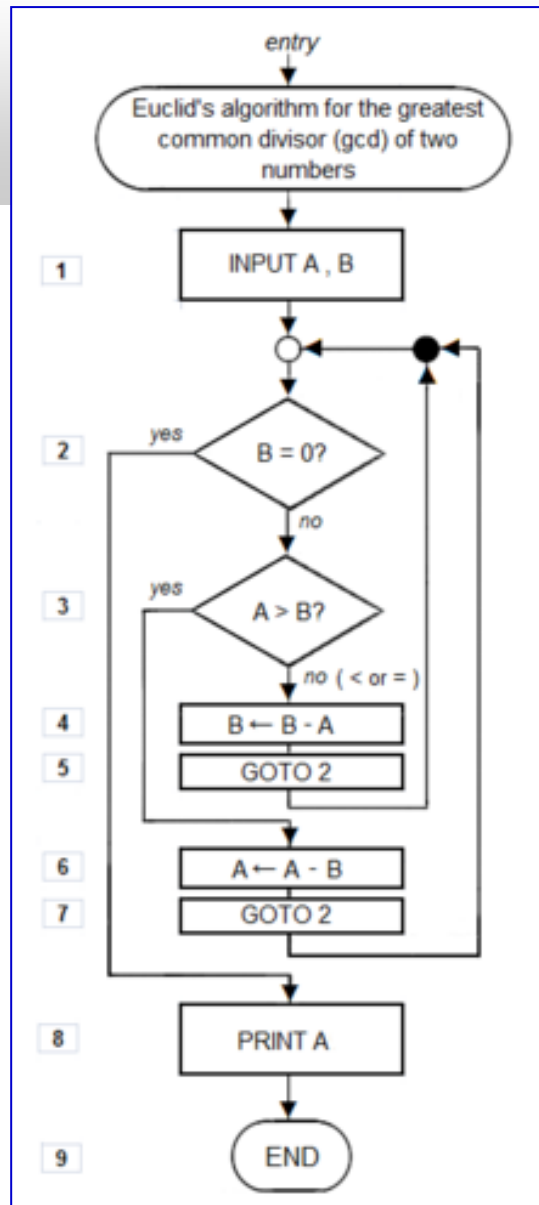
A collection of software to manage the hardware and to provide services to application programs.

OS examples: Microsoft Windows, Apple Mac OSX, Android, Linux.



Algorithm

Step-by-step procedure for calculation



Muḥammad ibn Mūsā al-Khwārizmī (Persian)



Programming languages

Generations

1. Machine code (100011 00011 01000...)

2. Assembly code →

3. Closer to human languages,
compiled or assembled prior to execution
(e.g. Java, C, Python, Pascal)

4. Domain-specific languages
(e.g. COBOL)

5. Problem solving by using program constraints,
instead of algorithms written by programmer
(artificial intelligence, learning from examples).

<u>Address</u>	<u>Instruction mnemonic</u>	<u>Arguments</u>
00000000	push	ebp
00000001	mov	ebp, esp
00000003	movzx	ecx, [ebp+arg_0]
00000007	pop	ebp
00000008	movzx	dx, cl
0000000C	lea	eax, [edx+edx]
0000000F	add	eax, edx
00000011	shl	eax, 2
00000014	add	eax, edx
00000016	shr	eax, 8
00000019	sub	cl, al
0000001B	shr	cl, 1
0000001D	add	al, cl
0000001F	shr	al, 5
00000022	movzx	eax, al
00000025	ret	



Python versus assembler example

Program for printing a „Hello World” message

X86 assembly language – x86-64 Linux, AT&T syntax

```
.section          .rodata
string:
    .ascii "Hello, World!\n\0"

length:
    .quad . -string          #Dot = 'here'

.section          .text
.globl _start           #Make entry point visible to linker
_start:
    movq $4, %rax        #4=write
    movq $1, %rbx        #1=stdout
    movq $string, %rcx
    movq length, %rdx
    int $0x80            #Call Operating System
    movq %rax, %rbx      #Make program return syscall exit status
    movq $1, %rax        #1=exit
    int $0x80            #Call System Again
```



Python versus assembler example

Program for printing a „Hello World” message

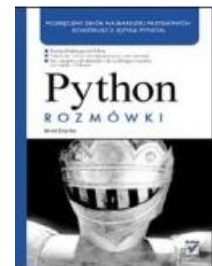
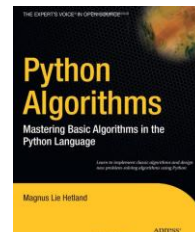
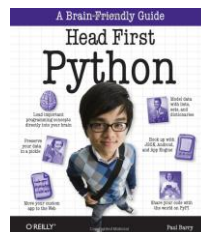
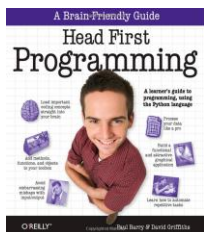
Code in Python language

```
print "Hello World"
```



Literature

- **Head First Programming: A Learner's Guide to Programming Using the Python Language** by David Griffiths
- **Head First Python** by Paul Barry
- **Python Algorithms: Mastering Basic Algorithms in the Python Language** by Magnus Lie hetland
- **Python. Rozmówki**, Brad Dyley
- **Python od Podstaw** – zespół autorów
- **Zanurkuj w pythonie** (http://pl.wikibooks.org/wiki/Zanurkuj_w_Pythonie)
- **Programowanie z Pythonem** podręcznik stworzony dla studentów I roku neuroinformatyki I fizyki medycznej na Wydziale Fizyki Uniwersytetu Warszawskiego (http://brain.fuw.edu.pl/edu/TI:Programowanie_z_Pythonem/Wersja_do_druku)
- Many many more books and tutorials available on the Internet





Final mark

- 5 computer tests during the semester (75 %)
- Activity during classes (attendance, presentations, programming project) (25 %)

Class participants are requested to bring their electronic student ID cards.



General Advice

- Slides shown during lectures do not cover all knowledge needed for passing the exam.
- Knowledge is acquired through studying and exercise.
- It is wise to take notes.
- It is worth browsing tutorials of Python modules used during classes.



Used Materials

1. Python for Scientist and Engineers – slides from course by Enthought, Inc. www.enthought.com
2. www.pl.python.org
 1. <http://pl.python.org/docs/>
 2. <http://pl.python.org/kursy,jezyka.html>
 3. <http://pl.python.org/wyklady.html>



News & links

1. SciPy conference with video presentations
http://conference.scipy.org/scipy2013/presentation_detail.php?id=129
2. Enthought supports development of distributed computing
<https://www.enthought.com/company/news/doe-sbir-grant>
3. Microsoft made a donation of \$100,000 to sponsor IPython's continued development
<http://ipython.org/microsoft-donation-2013.html>