Technical University of Lodz
Institute of Electronics

Algorithms and Data Structures

Introduction for Biomedical Engineering IFE Students

Łódź 2012

General Information

- Lecturers :

Andrzej Materka

andrzej.materka[at]p.lodz.pl
http://amaterka.pl
Phone: 42 631 26 26
Room: 322, third floor
Consultations: Monday

Marek Kociński

marek.kocinski[at]p.lodz.pl
http://www.eletel.p.lodz.pl/kocinski/
Phone: 42 631 36 38
Room: 205, second floor
Consultations: check in the callendar
https://poczta2011.p.lodz.pl/service/user/marek.kocinski@p.lodz.pl/studenci_2012.html

General Information

- **Lecture/Tutorial**: 30h (15 x 2 hours/week)
- **Venue**: room 413, building B9, Wólczańska 211/215
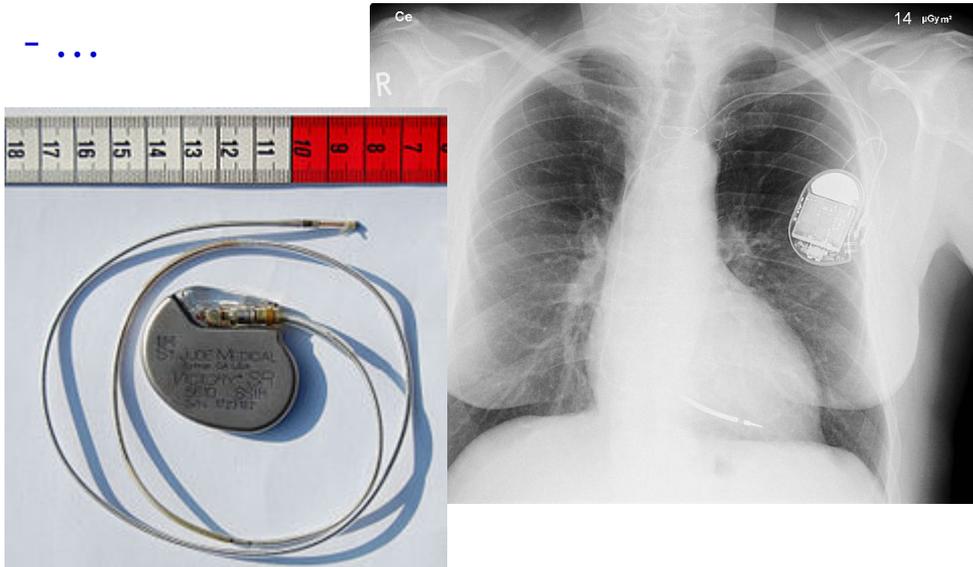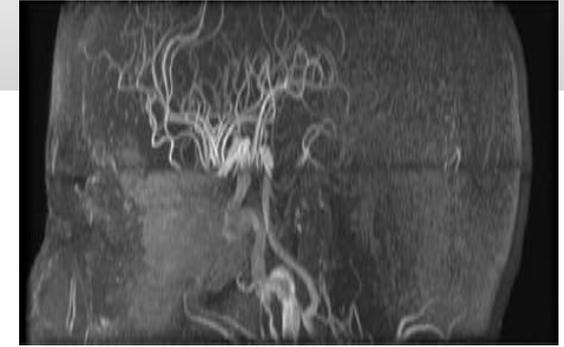
- **Credits:**   2 ECTS points (1 ECTS point = 25 - 30 h)
- **Self work:** 25 - 30 hours

- **Programming Language:**   Python
- **Tools:** Enthought Python Dystribution (EPD)

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

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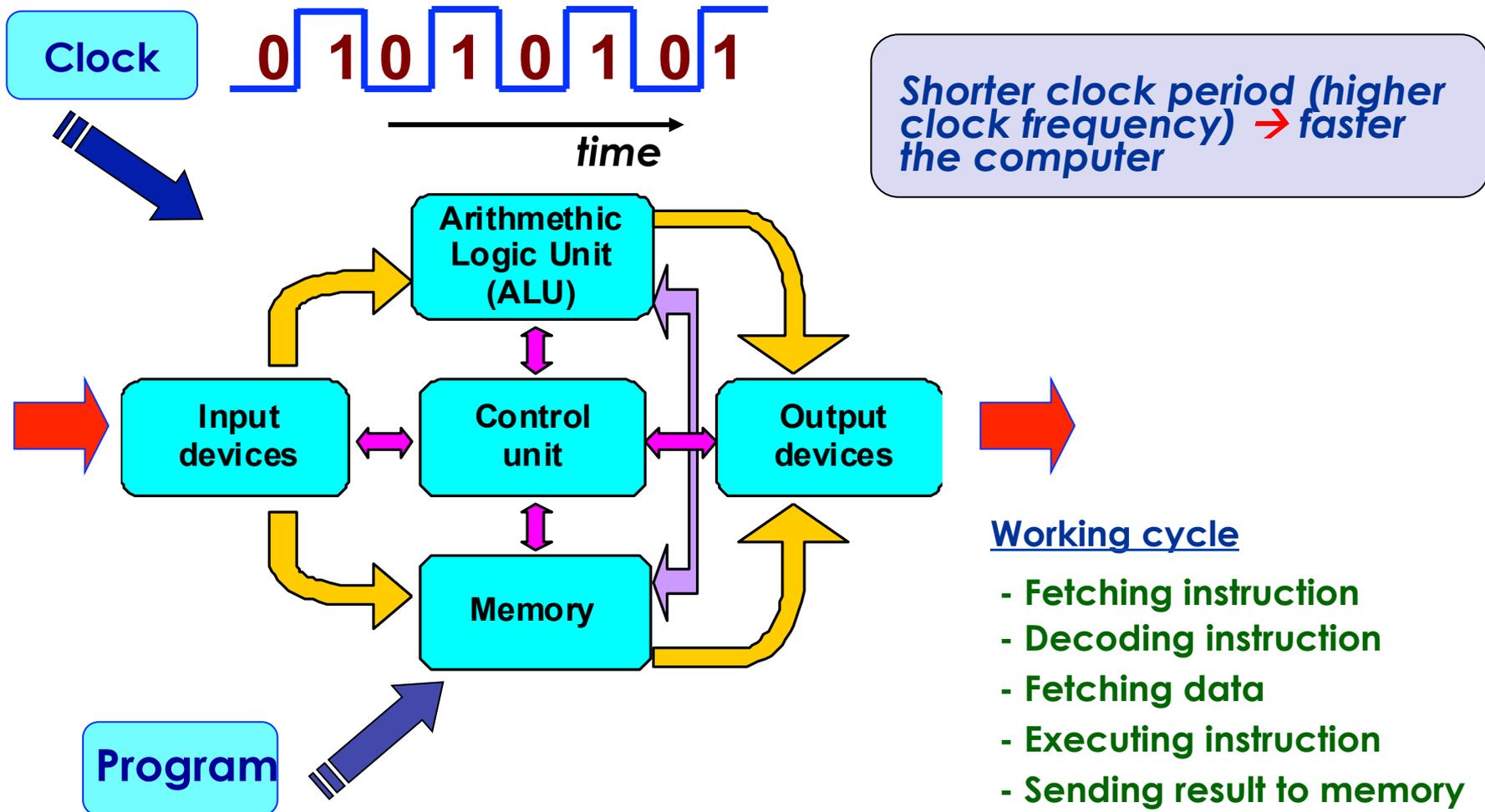- Medical Signal Analysis
- Surgery Navigation Aids
- Pacemakers
- …

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

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

Basics of Computer Architecture
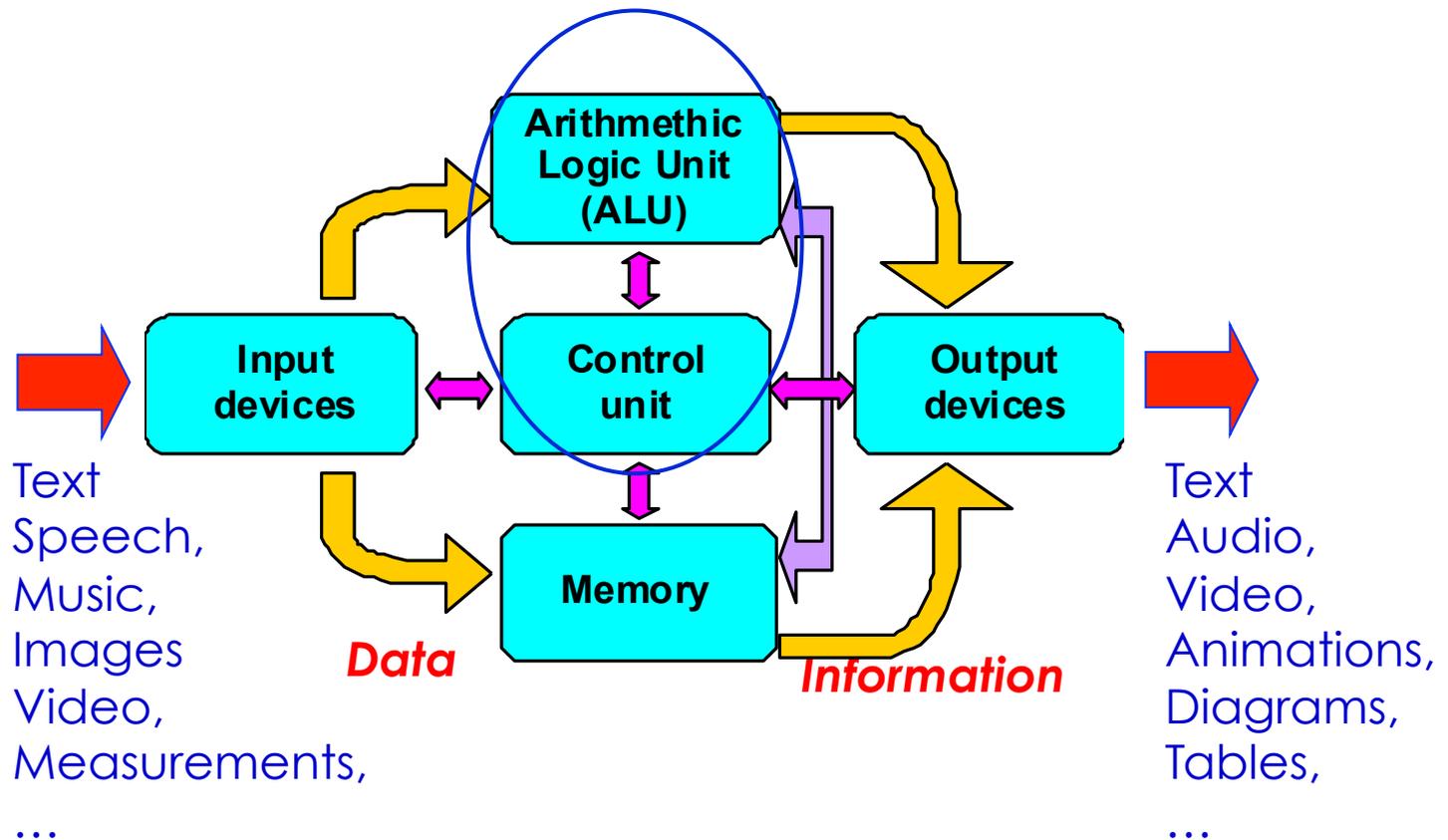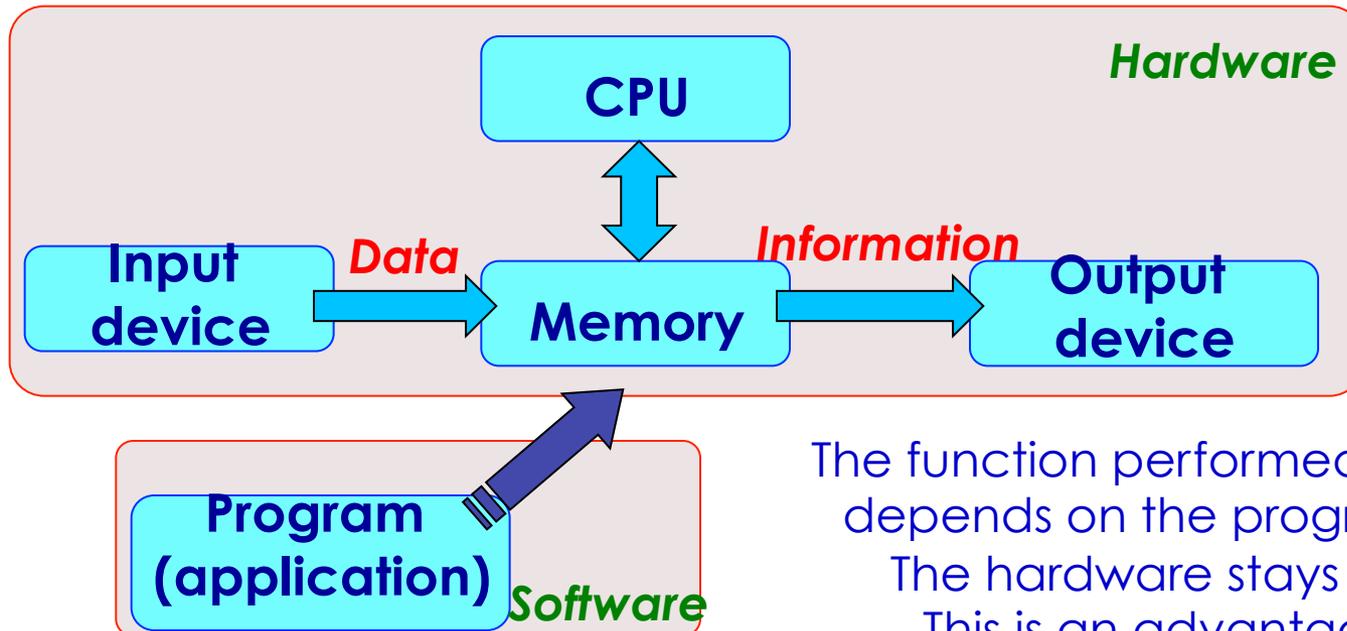
Clock

0 1 0 1 0 1 0 1

time

Shorter clock period (higher clock frequency) → faster the computer

Arithmethic Logic Unit (ALU)

Input devices

Control unit

Output devices

Memory

Program

Working cycle

- Fetching instruction
- Decoding instruction
- Fetching data
- Executing instruction
- Sending result to memory

Basics of Computing

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

Text
Speech,
Music,
Images
Video,
Measurements,

…

Data

Information

Text
Audio,
Video,
Animations,
Diagrams,
Tables,

…

The need for programming

CPU

Hardware

Input device

Data

Memory

Information

Output device

Program (application)

Software

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).

Operating system (OS)
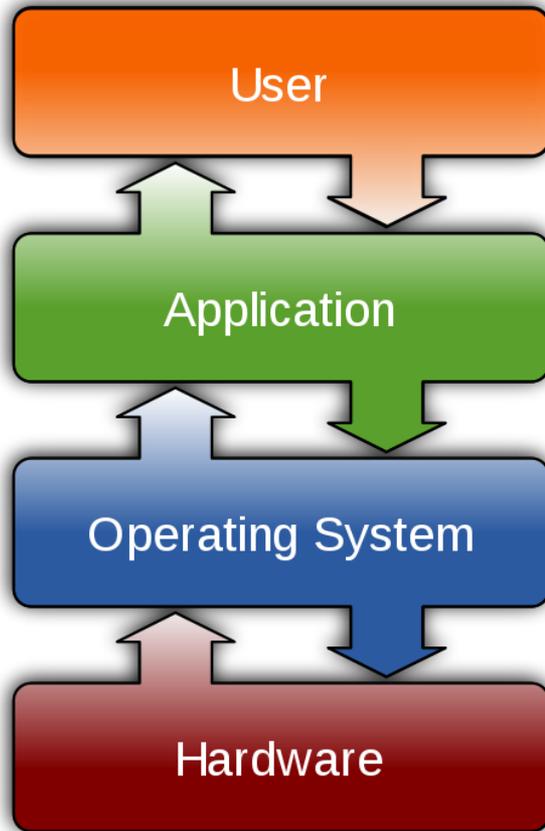
A collection of software to manage the hardware and to provide services to application programs.

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

Algorithm

entry

Euclid's algorithm for the greatest common divisor (gcd) of two numbers

1 INPUT A , B

2 yes B = 0?
   no

3 yes A > B?
   no (< or =)

4 B ← B - A
5 GOTO 2

6 A ← A - B
7 GOTO 2

8 PRINT A

9 END
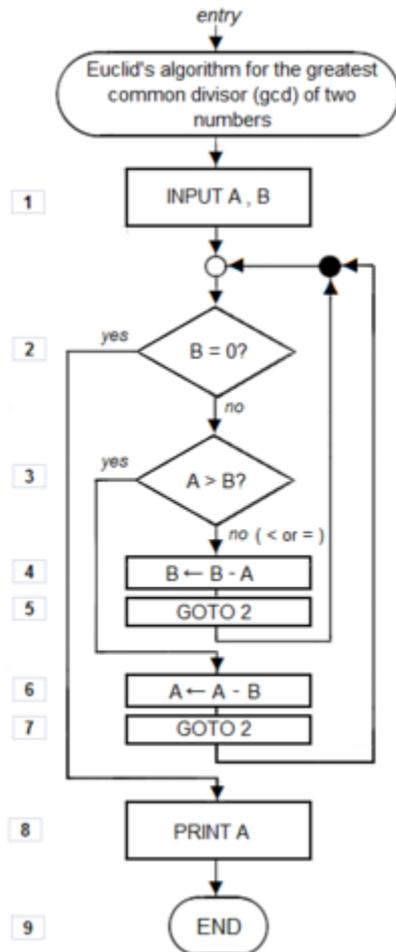
Muḥammad ibn Mūsā al-Khwārizmī (Persian)

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).

Address	Instruction mnemonic	Arguments
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	retn	

Python versus assembler example

Program for printing a „Hello World" message

```
 .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

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 Warszawskeigo (http://brain.fuw.edu.pl/edu/TI:Programowanie_z_Pythonem/Wersja_do_druku)
- Many many more books and tutorials available on the Internet
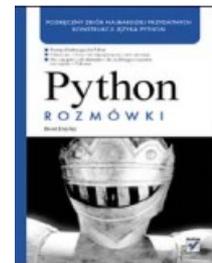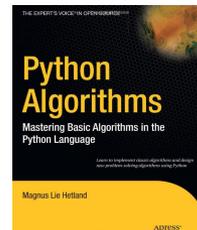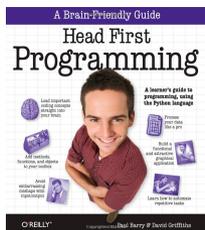
Final mark

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

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

Download and Installation of EPD

Download and Installation of EPD

Download and Installation of EPD