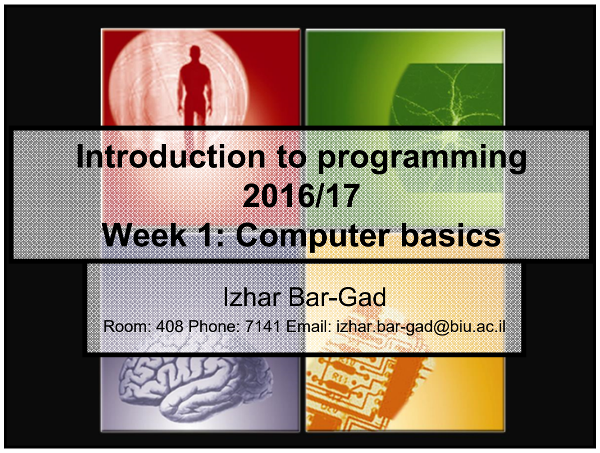
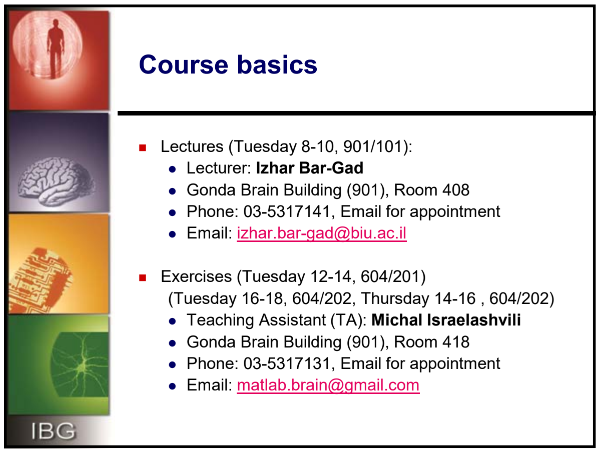


Introduction to programming 2016/17 Week 1: Computer basics

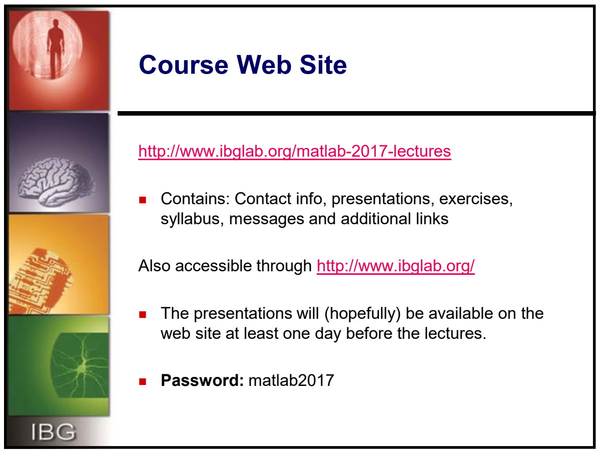
Izhar Bar-Gad
Room: 408 Phone: 7141 Email: izhar.bar-gad@biu.ac.il



Course basics

- Lectures (Tuesday 8-10, 901/101):
  - Lecturer: **Izhar Bar-Gad**
  - Gonda Brain Building (901), Room 408
  - Phone: 03-5317141, Email for appointment
  - Email: izhar.bar-gad@biu.ac.il
- Exercises (Tuesday 12-14, 604/201)
  (Tuesday 16-18, 604/202, Thursday 14-16 , 604/202)
  - Teaching Assistant (TA): **Michal Israelashvili**
  - Gonda Brain Building (901), Room 418
  - Phone: 03-5317131, Email for appointment
  - Email: matlab.brain@gmail.com

IBG



Course Web Site

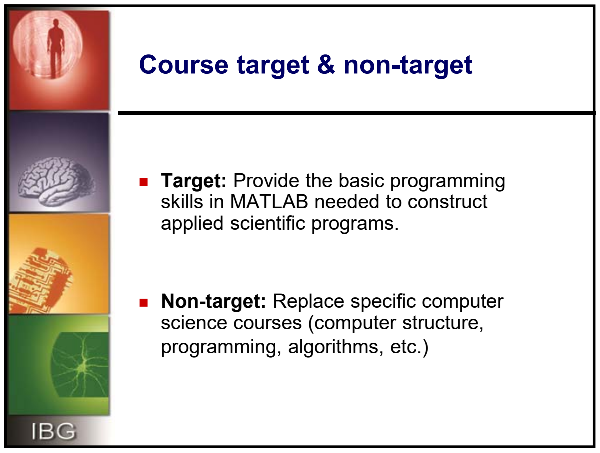
http://www.ibglab.org/matlab-2017-lectures

- Contains: Contact info, presentations, exercises, syllabus, messages and additional links

Also accessible through http://www.ibglab.org/

- The presentations will (hopefully) be available on the web site at least one day before the lectures.
- **Password:** matlab2017

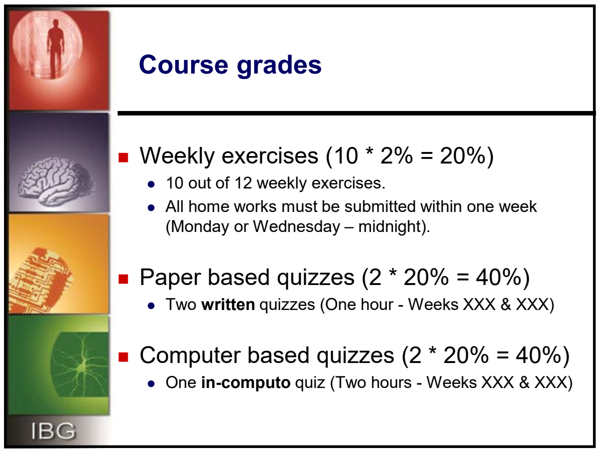
IBG



Course target & non-target

- **Target:** Provide the basic programming skills in MATLAB needed to construct applied scientific programs.
- **Non-target:** Replace specific computer science courses (computer structure, programming, algorithms, etc.)

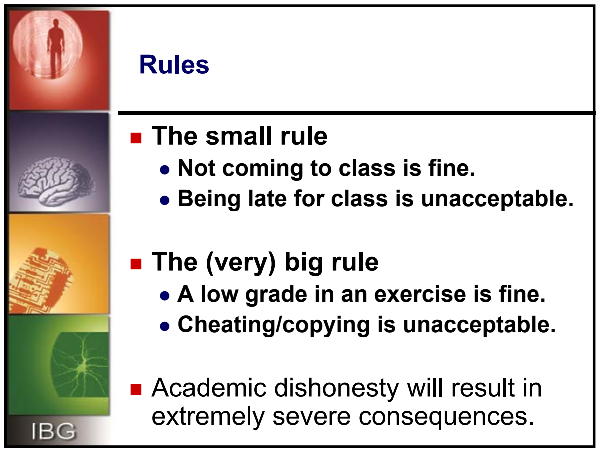
IBG



Course grades

- Weekly exercises (10 * 2% = 20%)
  - 10 out of 12 weekly exercises.
  - All home works must be submitted within one week (Monday or Wednesday – midnight).
- Paper based quizzes (2 * 20% = 40%)
  - Two **written** quizzes (One hour - Weeks XXX & XXX)
- Computer based quizzes (2 * 20% = 40%)
  - One **in-computo** quiz (Two hours - Weeks XXX & XXX)

IBG



Rules

- **The small rule**
  - **Not coming to class is fine.**
  - **Being late for class is unacceptable.**
- **The (very) big rule**
  - **A low grade in an exercise is fine.**
  - **Cheating/copying is unacceptable.**
- Academic dishonesty will result in extremely severe consequences.

IBG

Lecture 1 - Outline

- Computers
  - Hardware
  - Binary representation
- Programming
  - Software
  - Types of programs
  - Programming languages

Introduction to computers

- A computer is a device or machine for **processing** information from **data** according to a **program**. (Wikipedia)

The Processing Cycle I

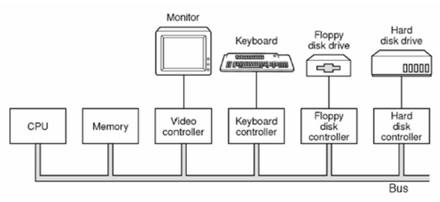
- Input comes in from somewhere
  - Keyboard, mouse, memory, camera, …
- The system does something/s with it
  - Add, subtract, move from place to place, …
- Output goes out to somewhere
  - Monitor, speaker, memory, robot, …

The Processing Cycle II

- Computer =
  input + processor + memory + output
- >99.9% of today’s computers are embedded (or hidden) in cars, TVs, microwaves, toys ...
- Of the remaining <0.1%, the PC is only one type and we will focus on it. However, there are others: smart phones, tablets, servers, batch processors, supercomputers, …

IBG

Computer Hardware - simple



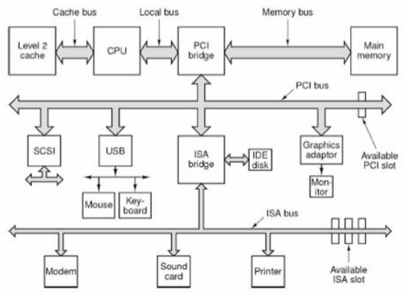
IBG

Computer hardware – real life

Let’s disassemble a computer…

IBG

Computer Hardware - complex



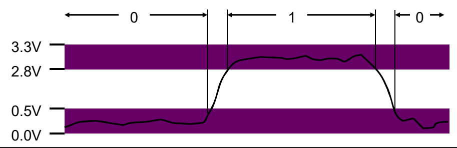
Information transfer and storage

So what is this loosely termed information, flowing in all those buses ?

(or using a better term: how is information encoded in the computer?)

Information encoding

- Humans use (mostly) base 10 numbers for calculation → digits.
- Computers use primarily bi-stable components, utilizing base 2 numbers → bits.



Information encoding - bases

- Decimal numbers - Base 10
  - 10 digits - 0, 1, …, 9
  - Of those we create numbers:
    0,1,2,…,9,10,11,12,…, 137, 138, …
  - Positional notation based on powers of 10.
    $1458 = 8*10^0 + 5*10^1 + 4*10^2 + 1*10^3$
- Binary Numbers – Base 2
  - 2 bits: 0, 1
  - Of those we create numbers:
    0,1,10,…, 1001,1010, 1011, 1100,…
  - Positional notation on the basis of powers of 2.
    $1101_2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3$

IBG

Binary and Decimal

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
…	…

IBG

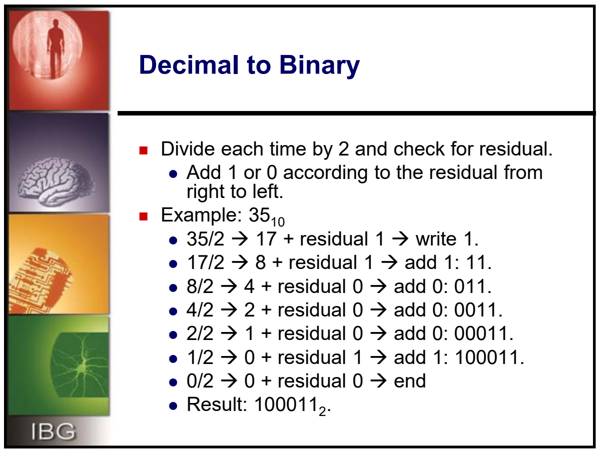
Converting bases

- Binary to Decimal

$1101_2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 = 13_{10}$

$100110_2 = 0*2^0 + 1*2^1 + 1*2^2 + 0*2^3 +$
$0*2^4 + 1*2^5 = 38_{10}$

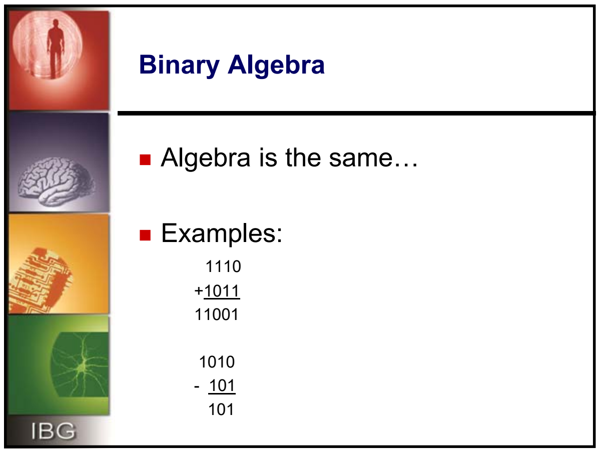
IBG



Decimal to Binary

- Divide each time by 2 and check for residual.
  - Add 1 or 0 according to the residual from right to left.
- Example: 35_{10}
  - 35/2 → 17 + residual 1 → write 1.
  - 17/2 → 8 + residual 1 → add 1: 11.
  - 8/2 → 4 + residual 0 → add 0: 011.
  - 4/2 → 2 + residual 0 → add 0: 0011.
  - 2/2 → 1 + residual 0 → add 0: 00011.
  - 1/2 → 0 + residual 1 → add 1: 100011.
  - 0/2 → 0 + residual 0 → end
  - Result: 100011_2.

IBG



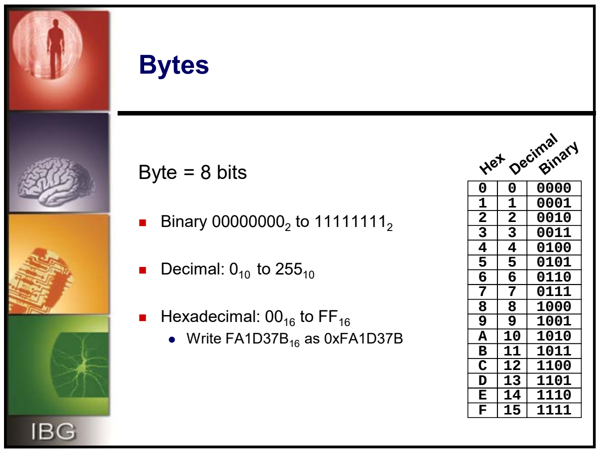
Binary Algebra

- Algebra is the same…
- Examples:

$$\begin{array}{r} 1110 \\ +\underline{1011} \\ 11001 \end{array}$$

$$\begin{array}{r} 1010 \\ -\ \underline{101} \\ 101 \end{array}$$

IBG



Bytes

Byte = 8 bits

- Binary 00000000_2 to 11111111_2
- Decimal: 0_{10} to 255_{10}
- Hexadecimal: 00_{16} to FF_{16}
  - Write $FA1D37B_{16}$ as 0xFA1D37B

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

IBG

Lecture 1 - Outline

- Computers
  - Hardware
  - Binary representation
- Programming
  - Software
  - Types of programs
  - Programming languages

IBG

Introduction to Programming I

- **Program** – Sequence of instructions that a computer can interpret and execute → written in a specific code.
  - Example: A problematic program to enter the Gonda Brain Research building:
    - Enter through the university gate.
    - Turn right.
    - Walk 30 meters.
    - Turn left.
    - Go up 5 stairs.
    - Enter the door.
- **Accuracy** is essential and so is **completeness**, otherwise errors will occur → **Bugs**

IBG

Introduction to Programming II

- Program **Work Flow**
  - Beginning – at a specific point.
  - Execution of commands.
  - End – at a specific point.
- A program can be very small (a few lines of code), or very large (Windows 8 contains ~50,000,000 lines of code), and can consist of many **files**.
- **Programming** – creating or changing a program**.**

IBG

Hardware / Software

- Hardware
  - Physical substrate which stores and executes the software.
- Software
  - All the information processed by computer system: programs and data (Alan Turing).
  - A computer program encoded in such a fashion that the program contents can be changed with minimal effort (John Tukey).

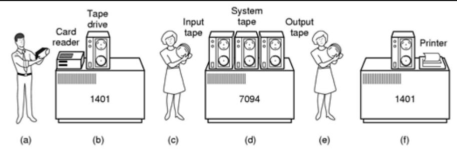
IBG

Operating System (OS)

- An operating system (OS) is the system software responsible for the direct control and management of hardware and basic system operations. Additionally, it provides a foundation upon which to run application software .
- It is an extended machine (top-down)
  - Hides the messy details which must be performed
  - Presents user with a virtual machine, easier to use
- It is a resource manager (bottom-up)
  - Each program gets time with the resource
  - Each program gets space on the resource

IBG

Early “human” OS



Early batch system
- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

IBG

Machine language

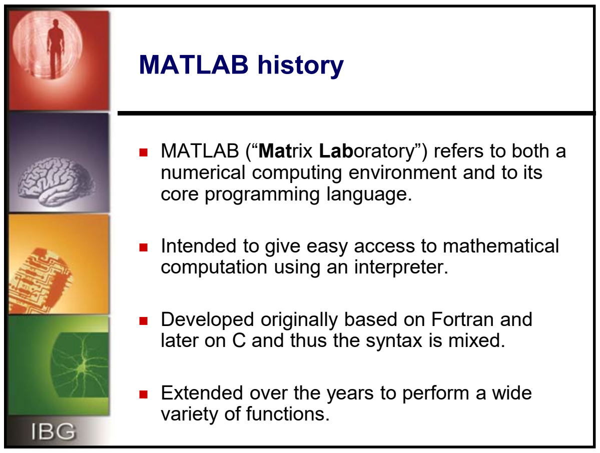
Address	Machine code (binary)	Machine code (hex)	Assembly
0	10101001	A9	LDA
1	00000010	02	#02
2	01101001	69	ADC
3	00000010	02	#02
4	10000101	85	STA
5	11001011	CB	$CB

Computer languages I

- The computer can understand **only** machine code.
- Thus, any instructions must be either converted to machine code **(compiler)** or run by another program running in machine code **(interpreter)**

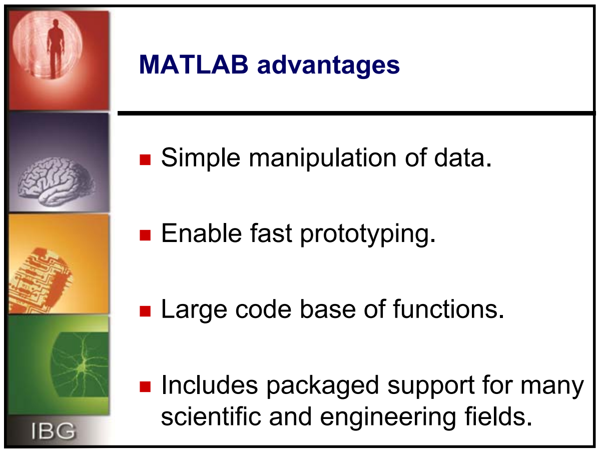
Computer languages II

- Many languages exist serving different purposes.
- C – built originally for system's programmers
- C++ - object oriented programming
- Java – portable code across platforms
- And last but not least **MATLAB**



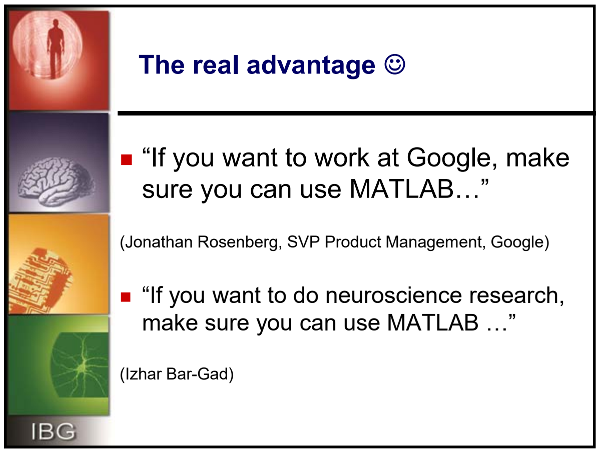
MATLAB history

- MATLAB (“**Mat**rix **Lab**oratory”) refers to both a numerical computing environment and to its core programming language.
- Intended to give easy access to mathematical computation using an interpreter.
- Developed originally based on Fortran and later on C and thus the syntax is mixed.
- Extended over the years to perform a wide variety of functions.



MATLAB advantages

- Simple manipulation of data.
- Enable fast prototyping.
- Large code base of functions.
- Includes packaged support for many scientific and engineering fields.



The real advantage ☺

- “If you want to work at Google, make sure you can use MATLAB…”

(Jonathan Rosenberg, SVP Product Management, Google)

- “If you want to do neuroscience research, make sure you can use MATLAB …”

(Izhar Bar-Gad)