

# BIL 111 Intro to Computer Engineering

Ammar Daskin

Course administrivia

Intro to CE

- What is computer engineering?

Problem Solving with Computers

- Computers as binary machines
- Binary numeral system
- Data representation
  - Numerics, floating points, images, sounds, etc.
- Computer Architecture
  - Data and code representation
  - Von Neumann Architecture
  - CPU
    - Arithmetic logic unit
      - Logical gates (AND, OR, etc)
- Algorithm design and problem solving
- Complexity of the algorithms
  - Efficient and non-efficient algorithms
  - Difficult and easy problems

# Derse İşleniři ve Katılım

Dersimiz haftada 2 ders her Perřembe 13:30da B517 de

Normalde tüm teorik derslere %70, tüm uygulamalı derslere %80 katılım zorunluluđu var

Ancak derste yoklama almayacađım! İsteyenler gelmeyebilirler.

Ders teorik genelde notlar üzerinden anlatacađım.

Ders notları classrooma atılır ancak bir çok Őeyi bulamayabilirsiniz.

- ❖ Ders esnasında önemli grdüğünüz kısımları not alınız.

# classroom.google.com

Dersle ilgili tüm duyurular

Ödevler

Sınav sonuçları ve cevapları

Ve tüm yazışmalar

Ödevlere **private comment** yazabiliyorsunuz.

Notifikasyonları düzenlemek için

Gmailde

Filte oluşturun: [Create rules to filter your emails - Gmail Help](#)

Kategorileri aktif edin: [Add or remove inbox categories in Gmail - Computer](#)

# Ödevler

5 civarı

Toplam notun %30u

Haftalık konulardan sorular veya uygulamalar

# Ödevlere yardım alma ve kopya

- Dersteki tüm ödevleri **tek başınıza sadece dersin hocasından** yardım alarak bitirmeniz beklenmektedir.
- Ödevlerin herhangi bir parçasını birbirinizle paylaşmanız kopya olarak nitelendirilmektedir.
- Eğer internet, kitap vb kaynaklardan faydalandıysanız bunları ödevlerde belirtmeniz zorunludur.
- Birbirinizle ödevleri tartışmanız önerilmektedir. Ancak bu tartışmalar ödevlerin beraber yapılması boyutuna gitmemelidir.

# Ofis saatleri

Dersten hemen sonra

Ödevlerde takıldığınız kısımları,

Derste anlamadığınız kısımları

Veya başka konuları sorabilirsiniz

# Derse Genel Bakış

Dersin haftalık sürecinde C, Python, ve SQL gibi programla dillerinde örnek yapılacak olup, yine web uygulamalarına yönelik HTML, CSS ve Javascript tanıtılacaktır. Dersin başarıyla bitirilmesi için haftalık derslerin takip edilmesi ve ödevlerin zamanında yapılması gerekmektedir.



# Konular

1. Giriş: Bilgisayarla problem çözme: bilgisayar mimarisi, data gösterimi, ikili sayı sistemi, program
2. Algoritma: Pseudocode ve Scratch
3. Scratch de örnek problemler
4. OS: Linux yükleme ve terminal komutları
5. Vim,gedit, emacs, etc
6. Pseudocode ve Mantıksal Operatörler
7. Akış şemaları-şartlar-döngüler
8. C ve Python da akış şemasından koda çevirme
9. Latex, Jupyter notebook
10. Python pyplot, version control
11. Web sayfası tasarımı (HTML, CSS)
12. Intro Javascript
13. Database ve SQL
14. Intro to security
15. Intro to computer networks
16. Yapay zeka ve uygulamaları
17. Bulut servisler
18. Benzetim ve uygulamaları
19. Optimizasyon ve uygulamaları (Büyük veri ve uygulamaları)

# Kitap ve Kaynaklar

Derste her hafta anlatılacaklar dersten önce veya bazen sonra classrooma yüklenecektir.

Ayrıca bir kitap yoktur.

# Notlandırma

- %30 Ödev
- %30 Vize
- %40 Final

# Intro to Computer Engineering

Who is an engineer?

# Engineer vs Scientist

## Engineer:

- A person who solves problems (or design solutions)
  - by using available scientific knowledge

## Scientist:

- Extends the available scientific knowledge by exploring/observing/experimenting nature
- Or introduces new theories that can be used to explain/observe/experiment nature

Who is a computer engineer?

# A computer engineer

Design/builds new computer systems that can be used to solve problems

Design/builds new tools that makes possible to use computer systems to solve problems

Design/builds new tools that improves available computer technologies

And many more



# Problem solving with computers

[lecture1.pdf](#) ile birlikte islendi

What is a computer?

How computer works?

- Von Neumann architecture

What is data?

Data representation:

- 1's complement
- 2's complement
- RGB images

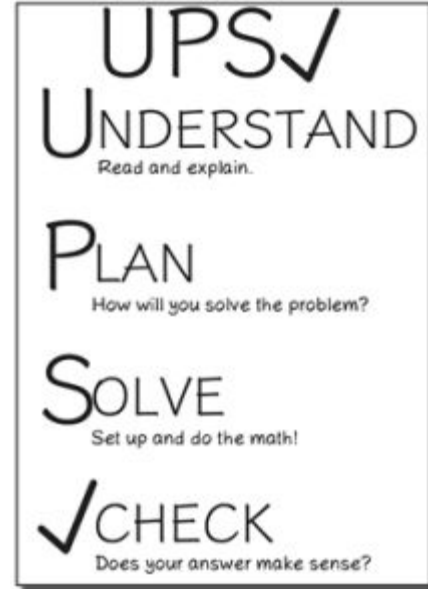
What is code?

How a program works?

# How to solve it?

## Polya's Problem-Solving Method

1. First, you have to *understand the problem*.<sup>[2]</sup>
2. After understanding, *make a plan*.<sup>[3]</sup>
3. *Carry out the plan*.<sup>[4]</sup>
4. *Look back* on your work.<sup>[5]</sup> How could it be better?



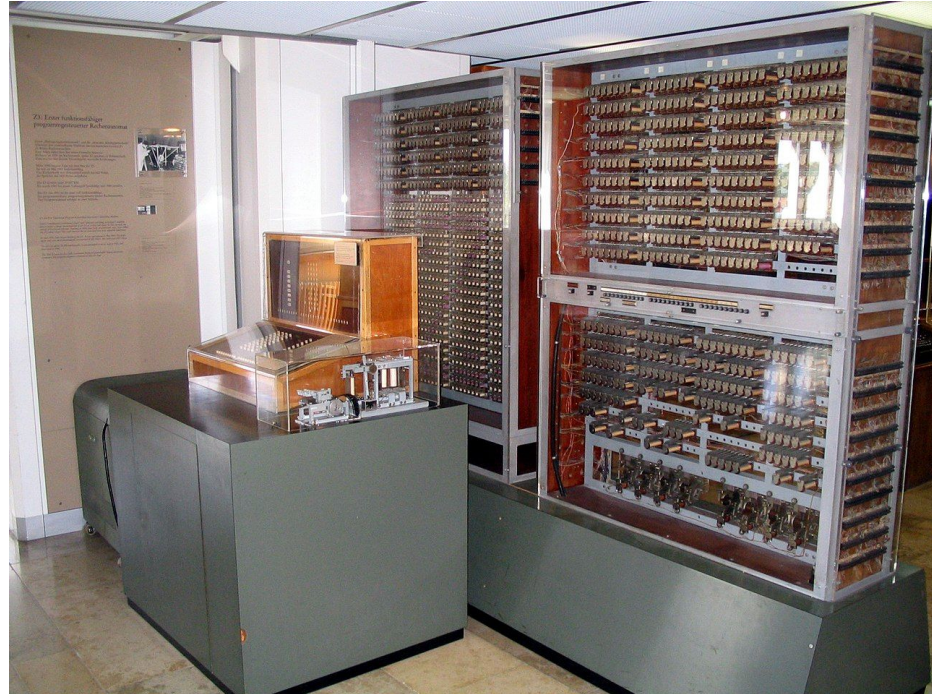
<https://www.opepp.org/lesson/hsdm-unit7-tool-for-field/>

[https://en.wikipedia.org/wiki/How\\_to\\_Solve\\_It](https://en.wikipedia.org/wiki/How_to_Solve_It)

How to use computers  
in problem solving?

# What is a computer?

Replica of [Konrad Zuse's Z3](#), 1st fully automatic, digital (electromechanical) computer



# What is a computer?

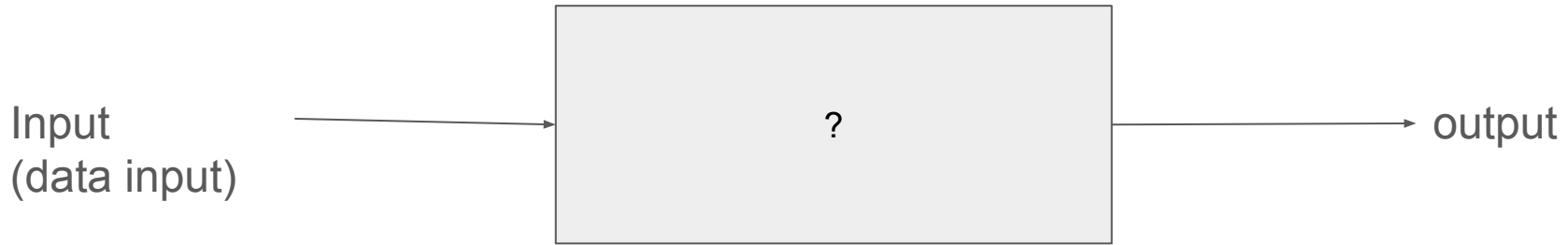


Modern computers are **digital** machines?

- Hardware: All the electrical components
  - Monitor, CPU, etc.
- Software: A computer runs a **program**
  - A game,
  - A movie program
  - Camera app

# How to solve a problem by a program?

Program yapısı



How to represent data?

- Characters, Numbers, Images, Sounds, Movies, Game Characters, etc.


# Symbols

# Numerical Digits

Arabic numerals (symbols) for writing numbers

0 1 2 3 4 5 6 7 8 9

Maya numeral systems

0	1	2	3	4
	•	••	•••	••••
5	6	7	8	9
—	• —	•• —	••• —	•••• —
10	11	12	13	14
==	• ==	•• ==	••• ==	•••• ==
15	16	17	18	19
===	• ===	•• ===	••• ===	•••• ===



# Binary (base 2) numeral system

Two symbols

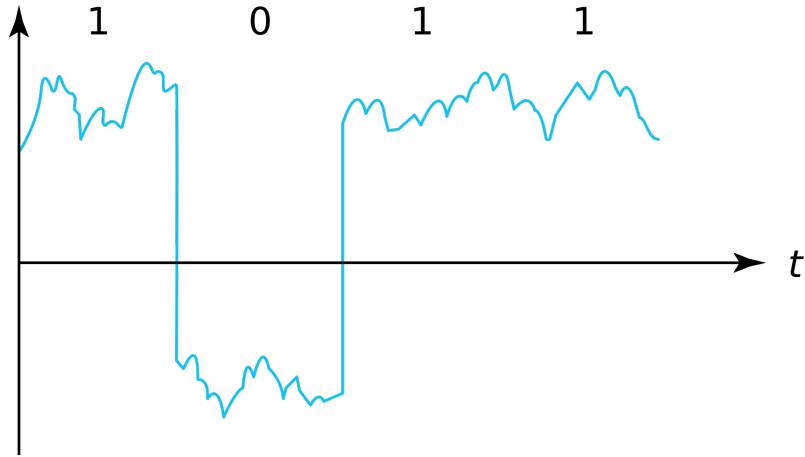
0

1

	Decimal number	Binary number		
	0	0	8	1000
	1	1	9	1001
	2	10	10	1010
	3	11	11	1011
	4	100	12	1100
	5	101	13	1101
	6	110	14	1110
	7	111	15	1111

# Computers are electrical machines

A digital signal



We use binary numeral system.

**Bit**

1 or 0

**Byte**

bit bit bit bit bit bit bit bit

00000010

Multiple-byte units						V•T•E
Decimal			Binary			
Value	Metric		Value	IEC	Memory	
1000	KB kilobyte		1024	KiB kibibyte	KB kilobyte	
1000 <sup>2</sup>	MB megabyte		1024 <sup>2</sup>	MiB mebibyte	MB megabyte	
1000 <sup>3</sup>	GB gigabyte		1024 <sup>3</sup>	GiB gibibyte	GB gigabyte	
1000 <sup>4</sup>	TB terabyte		1024 <sup>4</sup>	TiB tebibyte	TB terabyte	
1000 <sup>5</sup>	PB petabyte		1024 <sup>5</sup>	PiB pebibyte	-	
1000 <sup>6</sup>	EB exabyte		1024 <sup>6</sup>	EiB exbibyte	-	
1000 <sup>7</sup>	ZB zettabyte		1024 <sup>7</sup>	ZiB zebibyte	-	
1000 <sup>8</sup>	YB yottabyte		1024 <sup>8</sup>	YiB yobibyte	-	
1000 <sup>9</sup>	RB ronnabyte		1024 <sup>9</sup>	-	-	
1000 <sup>10</sup>	QB quettabyte		1024 <sup>10</sup>	-	-	

Orders of magnitude of data

[https://en.wikipedia.org/wiki/Byte#Multiple-byte\\_units](https://en.wikipedia.org/wiki/Byte#Multiple-byte_units)

Conversions from base- $x$  to base- $y$

# Decimal counting

100 10 1

000

100 10 1

001



100 10 1

002

100 10 1

003

100 10 1

004

100 10 1

005

100 10 1

006

100 10 1

007

100 10 1

008

100 10 1

009



100 10 1

010

100 10 1

123

$100 \times 1$  +  $10 \times 2$  +  $1 \times 3$

100 + 20 + 3

100 10 1

000

$10^2$     $10^1$     $10^0$

000

# Binary counting

$2^2$     $2^1$     $2^0$

000

4 2 1

000

4 2 1

001



4 2 1

010

4 2 1

011

4 2 1

100

4 2 1

101

4 2 1

110

4 2 1

111

# Binary counting

$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
16	8	4	2	1	
0	0	0	0	0	00

From base-2 to base-10

$$(10101110)_2 = (????)_{10}$$



From base-10 to base-2

$$(3456)_{10} = (????)_2$$

# Hexadecimal(base-16) numeral system

10-11-12-13-14-15

A-B-C-D-E-F

a-b-c-d-e-f

# Conversion from binary to hexadecimal

$$\begin{aligned}(1001011100)_2 &= 512_{10} + 64_{10} + 16_{10} + 8_{10} + 4_{10} \\ &= 604_{10}\end{aligned}$$

$$\begin{aligned}(1001011100)_2 &= 0010\ 0101\ 1100_2 \\ &= 2\ 5\ C_{16} \\ &= 25C_{16}\end{aligned}$$

Group of 2-quaternary(base-4)

$$25C_{16} = 021130_4$$

Group of 3-octal(base-8)

# Next week: How to represent data

Characters (symbols):

- 1,0, s, x\_,\$%^alsjkdom;lsmdf;l/\*65

Images ?

Musics ?

Numbers

- Integers
- Signed integers

Floating point numbers

- 1.4
- 5.25