CS 240
Computer Systems

#1: Course Introduction and Binary
Aug. 24, 2021 · Wade Fagen-Ulmschneider

Welcome to CS 240: Introduction to Computer Systems

Course Website: https://courses.engr.illinois.edu/cs240/

Description: Basics of computer systems. Number representations, assembly/machine language, abstract models of processors (fetch/execute, memory hierarchy), processes/process control, simple memory management, file I/O and directories, network programming, usage of cloud services. 3 credit hours.

Instructor:
Prof. Wade Fagen-Ulmschneider <waf@>, Course Instructor
Teaching Associate Professor of Computer Science, Grainger College of Engineering

Coursework and Grading

A total of 1,000 points are available in CS 240, along with many opportunities to earn extra credit. The points are broken down in the following way:

- **150 points**: Homeworks (10 × 15 points)
  - Points over 200 are extra credit!
  - Usually on PrairieLearn, but occasionally another platform
- **200 points**: Midterm Exams in CBTF (2 × 100 points)
  - Midterm 1 Exam (CBTF): Oct. 5 - Oct. 9
  - Midterm 2 Exam (CBTF: Nov. 30 - Dec. 4
- **500 points**: Machine Projects (10 weeks × 50 points)
  - Weekly machine problems, released every Friday and due the following Friday
  - Automatic 3-day grace period, extending to Monday
- **150 points**: Final Project
  - Multi-week Final Project, presented during finals weeks instead of a final exam (no final exam!)

We never curve individual exam or assignment scores; instead, if necessary, we may lower the points required for each grade cutoff to be lower than the stated cutoff. In no case will we raise the cutoff.

Final Course Grades

Your final course grade is determined by the number of points you earned during the semester:

Points	Grade	Points	Grade	Points	Grade
[1070, ∞)	A+	[930, 1070)	A	[900, 930)	A-
[870, 900)	B+	[830, 870)	B	[800, 830)	B-
[770, 800)	C+	[730, 770)	C	[700, 730)	C-
[670, 700)	D+	[630, 670)	D	[600, 630)	D-
		(600, 0]	F		

Foundations of Computer Systems

There are six major components to a computer, which we will refer to as the “foundations” of a computer system:

[1]:

[2]:

[3]:

[4]:

[5]:

[6]:

System-level Abstractions
After covering the “foundations”, we will begin to abstract the entire system as single **node** and explore more complex topics:

[1]:

[2]:

[3]:

Representing Data: Binary
All data within a computer is ______________; either **0** or **1**.

Converting between base-2 and base-10:

$1_2 = \qquad {}_{10}$

$10_2 = \qquad {}_{10}$

$11_2 = \qquad {}_{10}$

$100_2 = \qquad {}_{10}$

$101\ 1000_2 = \qquad {}_{10}$

Just like every digit has a “place value” in decimal (base-10), every digit has a “place value” in binary:

Binary Number:	0	1	0	1	1	0	0	0
(x) Place Value:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Place Value:	128	64	32	16	8	4	2	1
SUM:								

Any value can be represented in binary by writing it in base-2:

$4_{10} = \qquad {}_2$

$7_{10} = \qquad {}_2$

$18_{10} = \qquad {}_2$

In C/C++, you can write a number in binary by prefixing the number with **`0b`**:

$18_{10} =$ `0b`

$11_{10} =$ `0b`

$33_{10} =$ `0b`

Bit Manipulation:
We can manipulate bits by binary operations:

AND, **&** operator:

OR, **|** operator:

XOR, **^** operator:

NOT, **!** or **~** operator:

Bit Manipulation:

A	B	A & B	A | B	A ^ B	!A
1100	1010				
110011	11				
101	010				