No good party starts without introductions...
Wade Fagen-Ulmschneider (waf)
Teaching Associate Professor of Computer Science
Grainger College of Engineering
Nerding out in life...

Wade Fagen-Ulmschneider (waf)
Teaching Associate Professor of Computer Science
Grainger College of Engineering

Industry:
- Cisco
- Google
- Morgan Stanley

Creative Works:
- Gov. Inslee (WA)
- Gov. Beshear (KY)
- MOOC: Accel. CS Fund #1

Courses:
- CS 241
- CS 105
- CS 305
- CS 240
- CS 225
- STAT/CS/IS 107

MOOC: Accel. CS Fund.
Teaching Assistants

Eunice Zhou

Patrick Crain
Course Assistants

Bora Shim
Jeremy Shaffar
Jackson Kennel
Kevin Chen
You:
Overview: You Already Know
Overview: You Already Know

C++ Programming (CS 225)
Overview: **You Already Know**

C++ Programming (CS 225)

Data Structures (CS 225)
Overview: **You Already Know**

- C++ Programming (CS 225)
- Data Structures (CS 225)
- Algorithm Analysis (CS 173)
Overview: **You Already Know**

- C++ Programming (CS 225)
- Data Structures (CS 225)
- Algorithm Analysis (CS 173)
- Programming Skills (CS 125/126/225)
Overview: After CS 240
Overview: After CS 240

Foundational Computer Architecture
Overview: After CS 240

Foundational Computer Architecture

Operating System Design
Overview: After CS 240

- Foundational Computer Architecture
- Operating System Design
- Multiprogramming and Resource Sharing
Overview: After CS 240

Foundational Computer Architecture
Operating System Design
Multiprogramming and Resource Sharing
Cloud-based Infrastructure
Overview: After CS 240

Foundational Computer Architecture
Operating System Design
Multiprogramming and Resource Sharing
Cloud-based Infrastructure
Building Cloud-scale Applications
Course Structure
Course Structure

★ Lecture: Tuesday/Thursdays
Course Structure

★ Lecture: Tuesday/Thursdays

CS 225 (Spring 2018)

CS 240 (This Thursday, Lecture #2!)
Course Structure

★ Lecture: Tuesday/Thursdays

★ Weekly MPs and PL Homework
Course Structure

★ Lecture: Tuesday/Thursdays

★ Weekly MPs and PL Homework

★ Two Exams in the CBTF
Course Structure
★ Lecture: Tuesday/Thursdays
★ Weekly MPs and PL Homework
★ Two Exams in the CBTF
★ Final Course Project
Everything Else:

https://courses.grainger.illinois.edu/cs240/
Foundations of Computer Systems
Computer Systems Foundations
#1: Data
#2: Central Processing Unit
Computer Systems Foundations

#3: Memory and Storage
#4: Peripherals
Computer Systems Foundations

#5: Operating System
#6: Processes
System-Level Abstractions
System Level Abstractions
System Level Abstractions

#1: Virtual Machine
System Level Abstractions

#2: Containers
System Level Abstractions

#3: Nodes / Servers in the “Cloud”
Representing Data (Binary)
Representing Data

All data within a computer is:
$1_2 = 10$

$10_2 = 10$

$11_2 = 10$

$100_2 = 10$
\[
\begin{align*}
2_2 &= 10_2 \\
1_2 &= 10_2 \\
10_2 &= 10_2 \\
11_2 &= 10_2 \\
100_2 &= 10_2
\end{align*}
\]
101 \times 1000_2 = 10
Place Value of Digits

1  0  1  1  0  0  0  0

$2^6$  $2^5$  $2^4$  $2^3$  $2^2$  $2^1$  $2^0$
<table>
<thead>
<tr>
<th>Place Value of Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>64</td>
</tr>
</tbody>
</table>
Place Value of Digits

\[ \times \begin{array}{cccccccc}
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
64 & 32 & 16 & 8 & 4 & 2 & 1 & 10
\end{array} \]
### Place Value of Digits

<table>
<thead>
<tr>
<th></th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0</td>
<td>16</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ 10 \]
## Place Value of Digits

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

\[64 + 0 + 16 + 8 + 0 + 0 + 0 = 10\]
### Place Value of Digits

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

\[
64 + 0 + 16 + 8 + 0 + 0 + 0 + 0 + 10 = 88_{10}
\]
$$4_{10} = \begin{array}{c} 2 \end{array} = 0b$$

$$7_{10} = \begin{array}{c} 2 \end{array} = 0b$$

$$18_{10} = \begin{array}{c} 2 \end{array} = 0b$$
```c
#include <stdio.h>

int main() {
    int v1 = 0b10010;
    int v2 = 0b11001;
    int v3 = v1 + v2;
    printf("%d\n", v3);
    return 0;
}
```
Representing Data (Hexadecimal)
Binary Digits

Number of Students at Illinois:

\[ 0b\ 1100\ 1100\ 0110\ 1011 \]
Hexadecimal

Digits:
Place Value of Digits

0x c 0 f f f e e

$16^5 \quad 16^4 \quad 16^3 \quad 16^2 \quad 16^1 \quad 16^0$
Place Value of Digits

\[\begin{align*}
0 \times 16^5 & \quad 12 \times 16^5 \\
0 \times 16^4 & \quad 0 \times 16^4 \\
15 \times 16^3 & \quad 15 \times 16^3 \\
15 \times 16^2 & \quad 14 \times 16^2 \\
14 \times 16^1 & \quad 14 \times 16^1 \\
14 \times 16^0 & \quad 14 \times 16^0
\end{align*}\]
## Place Value of Digits

<table>
<thead>
<tr>
<th>Digit</th>
<th>Place Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$0 \times 16^5$</td>
<td>1048576</td>
</tr>
<tr>
<td>c</td>
<td>$\times 16^4$</td>
<td>65536</td>
</tr>
<tr>
<td>0</td>
<td>$15 \times 16^3$</td>
<td>4096</td>
</tr>
<tr>
<td>f</td>
<td>$15 \times 16^2$</td>
<td>256</td>
</tr>
<tr>
<td>f</td>
<td>$14 \times 16^1$</td>
<td>16</td>
</tr>
<tr>
<td>e</td>
<td>$14 \times 16^0$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12582912</td>
</tr>
</tbody>
</table>

- **12** multiplied by $16^5$ to give 12582912
Place Value of Digits

\[0 \times c \times 0 \times f \times f \times e \times e = 12,648,430_{10}\]
$11_{10} = 0x$  

$34_{10} = 0x$  

$87_{10} = 0x$  

$255_{10} = 0x$
<table>
<thead>
<tr>
<th>Decimal</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x1</td>
</tr>
<tr>
<td>2</td>
<td>0x2</td>
</tr>
<tr>
<td>3</td>
<td>0x3</td>
</tr>
<tr>
<td>4</td>
<td>0x4</td>
</tr>
<tr>
<td>5</td>
<td>0x5</td>
</tr>
<tr>
<td>6</td>
<td>0x6</td>
</tr>
<tr>
<td>7</td>
<td>0x7</td>
</tr>
<tr>
<td>8</td>
<td>0x8</td>
</tr>
<tr>
<td>9</td>
<td>0x9</td>
</tr>
<tr>
<td>10</td>
<td>0xa</td>
</tr>
<tr>
<td>11</td>
<td>0xb</td>
</tr>
<tr>
<td>12</td>
<td>0xc</td>
</tr>
<tr>
<td>13</td>
<td>0xd</td>
</tr>
<tr>
<td>14</td>
<td>0xe</td>
</tr>
<tr>
<td>15</td>
<td>0xf</td>
</tr>
</tbody>
</table>
Students at Illinois:

0b 1100 1100 0110 1011
People Following Tay on Twitter:

101 0100 1001 0010 1010 0110 0000
```c
#include <stdio.h>

int main() {
    int h1 = 0xc0ffee;
    int h2 = 0xf00d;
    printf("%x\n", h1 + h2);

    return 0;
}
```
Homework #1
PrairieLearn #01