CMPT 120: Introduction to Computing Science and Programming 1

Binary Encoding / Representation

Copyright © 2018, Liaqat Ali. Based on CMPT 120 Study Guide and Think Python - How to Think Like a Computer Scientist, mainly. Some content may have been adapted from earlier course offerings by Diana Cukierman, Anne Lavergn, and Angelica Lim. Copyrights © to respective instructors. Icons copyright © to their respective owners.
Reminders

Liaqat Ali, Summer 2018.
One-Stop Access To Course Information

- **Course website**: One-stop access to all course information.
  
  [http://www2.cs.sfu.ca/CourseCentral/120/liaqata/WebSite/index.html](http://www2.cs.sfu.ca/CourseCentral/120/liaqata/WebSite/index.html)
  
  - Course Outline
  - Exam Schedule
  - Python Info
  - [CourSys/Canvas](https://canvas.sfu.ca/courses/39187) link
  - Learning Outcomes
  - Office Hours
  - Textbook links
  - and more...
  - Grading Scheme
  - Lab/Tutorial Info
  - Assignments

- **Canvas**: Discussions forum - [https://canvas.sfu.ca/courses/39187](https://canvas.sfu.ca/courses/39187)

- **CourSys**: Assignments submission, grades - [www.coursys.sfu.ca](http://www.coursys.sfu.ca)

Liaqat Ali, Summer 2018.
How to Learn in This Course?

**A** Attend Lectures & Labs

**R** Read / review Textbook/Slides/Notes

**R** Reflect and ask Questions

**O** Organize — your learning activities on weekly basis, and finally...

**W** Write Code, Write Code, and Write Code.

Liaqat Ali, Summer 2018.
Deliverables

1. Deliverables are due by the given date and time.
2. For the course, we are using IDLE to write and run our Python code.
3. You can use the CSIL lab computers outside your lab hours.
4. Plan ahead your assignments and other deliverables. Computer crash, network problems etc. are not acceptable excuses for delays in deliverables.
5. You may use online Python interpreters for running and testing your codes, such as: https://repl.it/languages/Python3

Liaqat Ali, Summer 2018.
Labs

1. Each lab has an assigned TA.
2. Attend your assigned lab and show your work to your TA for the participation marks.
3. Class enrolments and lab swaps are closed now.
Course Topics

1. General introduction
2. Algorithms, flow charts and pseudocode
3. Procedural programming in Python
4. Data types and Control Structures
5. Fundamental algorithms
6. Binary encodings
7. Basics of computability and complexity
8. Basics of Recursion
9. Subject to time availability:
   ▫ Basics of Data File management
Today’s Topics

1. Data Representation (Binary Encoding)
   - ASCII
   - Unicode
   - Storage Units
   - Binary Addition
Data Representation

Liaqat Ali, Summer 2018.
In our common day life we use the Decimal Number System, only.

We write 11:
- it is understood that we talking about the decimal number system.
- We know it represents Eleven.
- But, what it 11 is a binary number?
- It then would mean 3.
- How can we can distingue number systems?
- One way is t use notation: \textit{Number}_{\text{base}}

So, write decimal 11 (base 10) as: 11_{10}
- 11_{10} thus represents \textit{ Eleven}.
- Base 10 position values: \begin{align*}
10^2 & \quad 10^1 & \quad 10^0 
\end{align*}

We write 11 in base 2 as: 11_{2}
- 11_{2} in Binary thus represents \textit{three}.
- Base 2 position values: \begin{align*}
2^2 & \quad 2^1 & \quad 2^0
\end{align*}

Similarly, we can write 11 in base 16 (hexadecimal) as: 11_{16}
- 11_{16} in hexadecimal means \textit{Seventeen}.
- Base 16 position values: \begin{align*}
16^2 & \quad 16^1 & \quad 16^0
\end{align*}
Storage Units

- **Bit**: storage to represent a binary 0 or 1.
- **Byte**: a group of 8-bits.
- More bigger storage units:

![Table]

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Symbol</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>(no prefix)</td>
<td>k</td>
<td>$2^0 = 1$</td>
</tr>
<tr>
<td>kilo-</td>
<td>k</td>
<td>$2^{10} = 1024 \approx 10^3$</td>
</tr>
<tr>
<td>mega-</td>
<td>M</td>
<td>$2^{20} = 1048576 \approx 10^6$</td>
</tr>
<tr>
<td>giga-</td>
<td>G</td>
<td>$2^{30} = 1073741824 \approx 10^9$</td>
</tr>
<tr>
<td>tera-</td>
<td>T</td>
<td>$2^{40} = 1099511627776 \approx 10^{12}$</td>
</tr>
</tbody>
</table>

- For example, “**12 megabytes**” is: $12 \times 2^{20}$ bytes = **12,582,912** bytes =>
  
  $12582912 \times 8$ bits = **100,663,296** bits.

Liaqat Ali, Summer 2018.
Binary Data Representation

• How data is stored inside computer when you write a statement: **marks = 12**
• The value **12** is an unsigned integer.
• Computer stores 12 in its binary equivalent form.
• 12 in binary is 1100.
• If computer uses 8 bits to store an unsigned integer, then 12 would be stored as:

<table>
<thead>
<tr>
<th>binary</th>
<th>decimal</th>
<th>binary</th>
<th>decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>15</td>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>1110</td>
<td>14</td>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>1101</td>
<td>13</td>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>1100</td>
<td>12</td>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>1011</td>
<td>11</td>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>1010</td>
<td>10</td>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>0000</td>
<td>0</td>
</tr>
</tbody>
</table>

For a positive integer represented by N binary digits the possible values are **0 <= value <= 2^N - 1**.
Representing signed Integers

- A **signed integer**: For a positive integer represented by $N$ binary digits, the possible values are $2^{N-1}-1 \leq \text{value} \leq 2^{N-1}-1$.

<table>
<thead>
<tr>
<th>Sign bit</th>
<th>N -1 Binary Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>+/-</td>
<td>$2^6$</td>
</tr>
<tr>
<td></td>
<td>64</td>
</tr>
</tbody>
</table>

- **+16**: 0 0 0 0 1 1 0 0
- **-16**: 1 0 0 0 1 1 0 0

Questions?