Variables, Basic I/O

CSCI 103L

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu

LECTURE 2
HTTP://WWW-SCF.USC.EDU/~CSCI103B
Outline

- Number Systems
- Text Systems
- Variables
- Input/Output
- Program
Number Bases

- There are many different number bases that are used for different reasons
  - Decimal (base 10)
    - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Binary (base 2)
    - 0, 1
  - Hexadecimal (base 16)
    - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - Octal
    - 0, 1, 2, 3, 4, 5, 6, 7

- Can you think of an example application that uses each number base?
Number Base Representations

- In base 10, each digit stands for a power of 10
  \[1364_{10} = 1 \times 10^3 + 3 \times 10^2 + 6 \times 10^1 + 4 \times 10^0\]
  \[= 1000 + 300 + 60 + 4\]

- In base 2, each bit stands for a power of 2
  \[1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\]
  \[= 8 + 0 + 0 + 1\]

- In base 16, each digit stands for a power of 16
  \[3D8 = 3 \times 16^2 + 13 \times 16^1 + 8 \times 16^0\]
  \[= 768 + 208 + 8\]
Converting from Base 10 to Base 2

- Convert \(201_{10}\) to binary

  \[
  \begin{align*}
  201 / 2 &= 100 \text{ R}1 \\
  100 / 2 &= 50 \text{ R}0 \\
  50 / 2 &= 25 \text{ R}0 \\
  25 / 2 &= 12 \text{ R}1 \\
  12 / 2 &= 6 \text{ R}0 \\
  6 / 2 &= 3 \text{ R}0 \\
  3 / 2 &= 1 \text{ R}1 \\
  1 / 2 &= 0 \text{ R}1
  \end{align*}
  \]

- Copy the remainders from the bottom to the top

  \[
  201_{10} = 11001001_2
  \]
Converting from Base 10 to Base 16

- Convert $199_{10}$ to hexadecimal
  
  $199 / 16 = 12$ R7
  
  $12 / 16 = 0$ R12
  
  Copy the remainders from the bottom to the top
  
  $199 = C7_{16}$

- Convert $4045_{10}$ to hexadecimal
  
  $4045 / 16 = 252$ R13
  
  $252 / 16 = 15$ R12
  
  $15 / 16 = 0$ R15
  
  Copy the remainders from the bottom to the top
  
  $4045_{10} = FCD_{16}$
Converting from Base 2 to Base 10

- Convert $10011_2$ to decimal
  
  $10011_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
  
  $= 16 + 0 + 0 + 2 + 1$
  
  $= 19_{10}$

- Convert $11001010_2$ to decimal
  
  $11001010_2 = 2^7 + 2^6 + 2^3 + 2^1$
  
  $= 128 + 64 + 8 + 2$
  
  $= 202_{10}$
Converting from Base 16 to Base 10

- Convert $3A7_{16}$ to decimal
  
  $3A7_{16} = 3 \times 16^2 + 10 \times 16^1 + 7 \times 16^0$
  
  $= 3 \times 256 + 10 \times 16 + 7 \times 1$
  
  $= 768 + 160 + 7$
  
  $= 935_{10}$

- Convert $ABCD_{16}$ to decimal
  
  $ABCD_{16} = 10 \times 16^3 + 11 \times 16^2 + 12 \times 16^1 + 13 \times 16^0$
  
  $= 10 \times 4096 + 11 \times 256 + 12 \times 16 + 13 \times 1$
  
  $= 40960 + 2816 + 192 + 13$
  
  $= 43981_{10}$
Converting from Base 16 to Base 2

- Convert C2D\textsubscript{16} to binary
  - It takes 4 bits to represent each hexadecimal digit
    - C\textsubscript{16} = 1100\textsubscript{2}
    - 2\textsubscript{16} = 0010\textsubscript{2}
    - D\textsubscript{16} = 1101\textsubscript{2}
    - C2D\textsubscript{16} = 1100 0010 1101\textsubscript{2}
Outline

- Number Systems
- Text Systems
- Variables
- Input/Output
- Program
ASCII Code

- Text characters are represented by a numeric value in computers
- ASCII (American Standard Code for Information Interchange)
  - 8 bits
  - Used primarily with the English character set

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
<th>Name</th>
<th>Char</th>
<th>Ctrl-char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Null</td>
<td>NUL</td>
<td>CTRL-@</td>
<td>32</td>
<td>20</td>
<td>Space</td>
<td>64</td>
<td>40</td>
<td>@</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Start of heading</td>
<td>SOH</td>
<td>CTRL-A</td>
<td>33</td>
<td>21</td>
<td>!</td>
<td>65</td>
<td>41</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Start of text</td>
<td>STX</td>
<td>CTRL-B</td>
<td>34</td>
<td>22</td>
<td>&quot;</td>
<td>66</td>
<td>42</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>End of text</td>
<td>ETX</td>
<td>CTRL-C</td>
<td>35</td>
<td>23</td>
<td>#</td>
<td>67</td>
<td>43</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>End of xmit</td>
<td>EOT</td>
<td>CTRL-D</td>
<td>36</td>
<td>24</td>
<td>$</td>
<td>68</td>
<td>44</td>
<td>D</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Enquiry</td>
<td>ENQ</td>
<td>CTRL-E</td>
<td>37</td>
<td>25</td>
<td>%</td>
<td>69</td>
<td>45</td>
<td>E</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Acknowledge</td>
<td>ACK</td>
<td>CTRL-F</td>
<td>38</td>
<td>26</td>
<td>&amp;</td>
<td>70</td>
<td>46</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Bell</td>
<td>BEL</td>
<td>CTRL-G</td>
<td>39</td>
<td>27</td>
<td>'</td>
<td>71</td>
<td>47</td>
<td>G</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Backspace</td>
<td>BS</td>
<td>CTRL-H</td>
<td>40</td>
<td>28</td>
<td>(</td>
<td>72</td>
<td>48</td>
<td>H</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>Horizontal tab</td>
<td>VT</td>
<td>CTRL-I</td>
<td>41</td>
<td>29</td>
<td>)</td>
<td>73</td>
<td>49</td>
<td>I</td>
</tr>
<tr>
<td>10</td>
<td>0A</td>
<td>Form feed</td>
<td>FF</td>
<td>CTRL-J</td>
<td>42</td>
<td>2A</td>
<td>*</td>
<td>74</td>
<td>4A</td>
<td>J</td>
</tr>
<tr>
<td>11</td>
<td>0B</td>
<td>Carriage feed</td>
<td>CR</td>
<td>CTRL-K</td>
<td>43</td>
<td>2B</td>
<td>+</td>
<td>75</td>
<td>4B</td>
<td>K</td>
</tr>
<tr>
<td>12</td>
<td>0C</td>
<td>Shift out</td>
<td>SO</td>
<td>CTRL-L</td>
<td>44</td>
<td>2C</td>
<td>,</td>
<td>76</td>
<td>4C</td>
<td>L</td>
</tr>
<tr>
<td>13</td>
<td>0D</td>
<td>Shift in</td>
<td>SI</td>
<td>CTRL-M</td>
<td>45</td>
<td>2D</td>
<td>.</td>
<td>77</td>
<td>4D</td>
<td>M</td>
</tr>
<tr>
<td>14</td>
<td>0E</td>
<td>Data line escape</td>
<td>DLE</td>
<td>CTRL-N</td>
<td>46</td>
<td>2E</td>
<td>;</td>
<td>78</td>
<td>4E</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td>0F</td>
<td>Device control 1</td>
<td>DC1</td>
<td>CTRL-P</td>
<td>47</td>
<td>2F</td>
<td>/</td>
<td>79</td>
<td>4F</td>
<td>O</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>Device control 2</td>
<td>DC2</td>
<td>CTRL-Q</td>
<td>48</td>
<td>30</td>
<td>0</td>
<td>80</td>
<td>50</td>
<td>P</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>Device control 3</td>
<td>DC3</td>
<td>CTRL-R</td>
<td>49</td>
<td>31</td>
<td>1</td>
<td>81</td>
<td>51</td>
<td>Q</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>Cancel</td>
<td>CAN</td>
<td>CTRL-S</td>
<td>50</td>
<td>32</td>
<td>2</td>
<td>82</td>
<td>52</td>
<td>R</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>End of medium</td>
<td>EM</td>
<td>CTRL-T</td>
<td>51</td>
<td>33</td>
<td>3</td>
<td>83</td>
<td>53</td>
<td>S</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>End of xmit block</td>
<td>EOT</td>
<td>CTRL-U</td>
<td>52</td>
<td>34</td>
<td>4</td>
<td>84</td>
<td>54</td>
<td>T</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>Escape</td>
<td>ESC</td>
<td>CTRL-V</td>
<td>53</td>
<td>35</td>
<td>5</td>
<td>85</td>
<td>55</td>
<td>U</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>Synchronous idle</td>
<td>SYN</td>
<td>CTRL-W</td>
<td>54</td>
<td>36</td>
<td>6</td>
<td>86</td>
<td>56</td>
<td>V</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>Bell</td>
<td>BEL</td>
<td>CTRL-X</td>
<td>55</td>
<td>37</td>
<td>7</td>
<td>87</td>
<td>57</td>
<td>W</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>Cancel</td>
<td>CAN</td>
<td>CTRL-Y</td>
<td>56</td>
<td>38</td>
<td>8</td>
<td>88</td>
<td>58</td>
<td>X</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>Start of text</td>
<td>STX</td>
<td>CTRL-Z</td>
<td>57</td>
<td>39</td>
<td>9</td>
<td>89</td>
<td>59</td>
<td>Y</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>Substitute</td>
<td>SUB</td>
<td>CTRL-A</td>
<td>58</td>
<td>3A</td>
<td>:</td>
<td>90</td>
<td>5A</td>
<td>Z</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>Escape</td>
<td>ESC</td>
<td>CTRL-B</td>
<td>59</td>
<td>3B</td>
<td>;</td>
<td>91</td>
<td>5B</td>
<td>[</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>File separator</td>
<td>FS</td>
<td>CTRL-C</td>
<td>60</td>
<td>3C</td>
<td>&lt;</td>
<td>92</td>
<td>5C</td>
<td>\</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>Group separator</td>
<td>GS</td>
<td>CTRL-D</td>
<td>61</td>
<td>3D</td>
<td>=</td>
<td>93</td>
<td>5D</td>
<td>]</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>Record separator</td>
<td>RS</td>
<td>CTRL-E</td>
<td>62</td>
<td>3E</td>
<td>&gt;</td>
<td>94</td>
<td>5E</td>
<td>^</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>Unit separator</td>
<td>US</td>
<td>CTRL-F</td>
<td>63</td>
<td>3F</td>
<td>?</td>
<td>95</td>
<td>5F</td>
<td>`</td>
</tr>
</tbody>
</table>

Image from http://www.commfront.com/images/Standard-ASCII-Table.jpg
Unicode

- Unicode contains more than 110,000 characters used in most of the world’s writing systems
  - 16 bits
Outline

- Number Systems
- Text Systems
- Variables
- Input/Output
- Program
### Variable Types

<table>
<thead>
<tr>
<th>Variable Type</th>
<th># bits</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>8</td>
<td>-2^8 to 2^8 – 1</td>
</tr>
<tr>
<td>short</td>
<td>16</td>
<td>-2^16 to 2^16 – 1</td>
</tr>
<tr>
<td>int</td>
<td>16</td>
<td>-2^16 to 2^16 – 1</td>
</tr>
<tr>
<td>long</td>
<td>32</td>
<td>-2^32 to 2^32 – 1</td>
</tr>
<tr>
<td>long long</td>
<td>64</td>
<td>-2^64 to 2^64 – 1</td>
</tr>
<tr>
<td>float</td>
<td>32</td>
<td>7 significant digits * 10^+/-38</td>
</tr>
<tr>
<td>double</td>
<td>64</td>
<td>16 significant digits * 10^+/-308</td>
</tr>
</tbody>
</table>

- The integer types can be unsigned, which will add one bit to the range, though it will always be positive.
C Variables

- A variable in a program is like a variable in math – it can take on different values at different times
- To add the number 23 to 45 and store it in a variable named “x”, write
  
  ```c
  int x;
  x = 23 + 45;
  ```
- Variables can also be added together, such as
  
  ```c
  int x, y, z;
  y = 23;
  z = 45;
  x = y + z;
  ```
- Then to print the value of x, write
  
  ```c
  printf ("x = %d\n", x);
  ```
- Or format it a little bit as
  
  ```c
  printf ("23 + 45 = %d\n", x);
  ```
- Print multiple variables
  
  ```c
  printf ("%d + %d = %d\n", y, z, x);
  ```
Arithmetic

- **Addition**
  ```c
  int x = 34 + 12;
  ```
- **Subtraction**
  ```c
  short y = 34 - 12;
  ```
- **Multiplication**
  ```c
  int z = 34 * 12;
  ```
- **Division**
  ```c
  int a = 34 / 12;
  ```
- **Modulus**
  ```c
  short b = 34 % 12;
  ```
- **Parentheses can be used to enforce order of operations**
  ```c
  long c = 34 * 12 + 13;
  long d = 34 * (12 + 13);
  ```
  › Note: Only parentheses can be used for changing the order of operations. Brackets and curly braces have different meanings in C.
- **Increment and Decrement**
  ```c
  int e = 3;
  e++;
  e--;  ```
Outline

- Number Systems
- Text Systems
- Variables
- Input/Output
- Program
Output to the Command Line

- To output to the command line, use the `printf` function
  
  ```
  printf("Hello CSCI 103");
  ```

- To output a new line, include a `\n` in the string
  
  ```
  printf("Hello CSCI\n103");
  ```
Escape Characters

- When printing values in a string in C, there are a few special characters:
  - \n – new line
  - \t – horizontal tab
  - \a – alert
  - \\ – backslash
  - \" – double quote

- What does the following line output?

```c
printf("Hello\n\nCSCI\n"103\n\\\n");
```
Printing Variables

- To output a variable in a `printf` statement, a variable code is required.

```
int num = 3;
printf ("The value is %d\n", num);
```

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Variable Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>%c</td>
</tr>
<tr>
<td>short</td>
<td>%d</td>
</tr>
<tr>
<td>int</td>
<td>%d or %i (signed)</td>
</tr>
<tr>
<td></td>
<td>%u (unsigned)</td>
</tr>
<tr>
<td></td>
<td>%o (unsigned octal value)</td>
</tr>
<tr>
<td></td>
<td>%x (unsigned hexadecimal value)</td>
</tr>
<tr>
<td>long</td>
<td>%d</td>
</tr>
<tr>
<td>float</td>
<td>%f (fixed-point notation)</td>
</tr>
<tr>
<td></td>
<td>%e (exponential notation)</td>
</tr>
<tr>
<td>double</td>
<td>%lf</td>
</tr>
<tr>
<td>string</td>
<td>%s</td>
</tr>
</tbody>
</table>
Formatting Output

- Precision matters when outputting certain values
  
  ```
  float num = 192.13f;
  printf("%f\n", num); // outputs 192.130000
  ```

- If you want to output with just two decimal places, you can use `%.2f`
  
  ```
  float num = 192.13f;
  printf("%.2f\n", num); // outputs 192.13
  ```
Reading Input

- To read input from a user, use the `scanf` function
  - The `scanf` function takes two parameters – the type of variable being read and the variable into which to read the data
  - You must have the `&` before the name of the variable into which you are reading the data.
    - We will learn why this is the case in a future lecture

```c
int num;
printf("Enter a number: ");
scanf("%d", &num);
```
Comments

- Comments allow a programmer to freely type without the compiler looking at it
  - Comments are used by programmers to explain what lines of code or sections of code do

- There are two types of comments in C
  - Single line comments
    ```c
    // here is a single line comment
    ```
  - Multi-line comments
    ```c
    /*
     * here is a multi-line comment
     */
    ```
Outline

- Number Systems
- Text Systems
- Variables
- Input/Output
- Program
First Program

- If you want to do any input or output using `scanf` or `printf`, you have to include the `stdio.h` header file in your C program.
- The entry point for every C/C++ program is the `main` function.
  - All executable code goes in the `main` function or in another function that is directly or indirectly called from the `main` function.

```c
#include <stdio.h>
void main() {
    printf("Hello CSCI 103\n");
}
```
Compiling

- After you write your code, you need to compile it down to the language your computer will understand
  > For C/C++, this is the native binary language of your operating system

- Many different compilers exist, running on the command line or through an integrated development environment (IDE)
Program

- Write a program that outputs a table with the number, the square, and the cube of the number. Here is the output.

\begin{verbatim}
c:\> MultTable.exe
Number Square Cube
------ ------ ----
  0    0    0
  1    1    1
  2    4    8
  3    9   27
  4   16   64
  5   25  125

c:\>
\end{verbatim}