CS 110 SI – Winter 2016

Session 2: Data Types, Math and User Input
Session Outline

1. Quick Review
   1.1 Data Types
   1.2 Math Operations
   1.3 Getting Input from the User
2. Examples
3. TIWIKWITCS110 – The art of naming
This week in a nutshell

- Data Types
- Variables and Constants
- Getting data from the user
- Doing things that are actually useful!
Data Types in C++

- Data types represent real world concepts
  - Whole numbers
  - Decimal numbers
  - Letters
  - Words
  - Logic
- However, they are just approximations using bits.
Integers

● Integers represent whole numbers
  - Used for counting, indices
  - In C++ uses the keyword “int”
  - You probably recognize it from “int main ()”

● Example declaration
  ```c
  int numberOfDaysInApril = 30;
  ```
Int Modifiers

• The number of bits can be modified using the following keywords:
  short, long, long long, unsigned

• These typically give you more bits if you need them, or assign the bits different meanings.
  Eg. long long int billGatesBankAccount;
Floating Point

- Floats represent numbers with decimals
  - Use any time you need really small (or large) numbers
  - C++ keywords: float, double
  - Double has more bits

- Example declaration:
  ```
  float massOfTheEarth;
  ```
Char

• Represents a letter
  – Actually stores a very small number (0 ... 256)
  – Looks up the characters value in the ASCII Table
  – Single quotes around a literal

• Example:
  
  char middleInitial = 'm';
<table>
<thead>
<tr>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NUL</td>
<td>32</td>
<td>Space</td>
<td>64</td>
<td>@</td>
<td>96</td>
<td>`</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SOH</td>
<td>33</td>
<td>!</td>
<td>65</td>
<td>A</td>
<td>97</td>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>STX</td>
<td>34</td>
<td>“</td>
<td>66</td>
<td>B</td>
<td>98</td>
<td>b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ETX</td>
<td>35</td>
<td>#</td>
<td>67</td>
<td>C</td>
<td>99</td>
<td>c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>EOT</td>
<td>36</td>
<td>$</td>
<td>68</td>
<td>D</td>
<td>100</td>
<td>d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ENQ</td>
<td>37</td>
<td>%</td>
<td>69</td>
<td>E</td>
<td>101</td>
<td>e</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ACK</td>
<td>38</td>
<td>&amp;</td>
<td>70</td>
<td>F</td>
<td>102</td>
<td>f</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BEL</td>
<td>39</td>
<td>'</td>
<td>71</td>
<td>G</td>
<td>103</td>
<td>g</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>BS</td>
<td>40</td>
<td>(</td>
<td>72</td>
<td>H</td>
<td>104</td>
<td>h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TAB</td>
<td>41</td>
<td>)</td>
<td>73</td>
<td>I</td>
<td>105</td>
<td>i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>LF</td>
<td>42</td>
<td>*</td>
<td>74</td>
<td>J</td>
<td>106</td>
<td>j</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>VT</td>
<td>43</td>
<td>+</td>
<td>75</td>
<td>K</td>
<td>107</td>
<td>k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>FF</td>
<td>44</td>
<td>,</td>
<td>76</td>
<td>L</td>
<td>108</td>
<td>l</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CR</td>
<td>45</td>
<td>-</td>
<td>77</td>
<td>M</td>
<td>109</td>
<td>m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>SO</td>
<td>46</td>
<td>.</td>
<td>78</td>
<td>N</td>
<td>110</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SI</td>
<td>47</td>
<td>/</td>
<td>79</td>
<td>O</td>
<td>111</td>
<td>o</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>DLE</td>
<td>48</td>
<td>0</td>
<td>80</td>
<td>P</td>
<td>112</td>
<td>p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>DC1</td>
<td>49</td>
<td>1</td>
<td>81</td>
<td>Q</td>
<td>113</td>
<td>q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>DC2</td>
<td>50</td>
<td>2</td>
<td>82</td>
<td>R</td>
<td>114</td>
<td>r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>DC3</td>
<td>51</td>
<td>3</td>
<td>83</td>
<td>S</td>
<td>115</td>
<td>s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>DC4</td>
<td>52</td>
<td>4</td>
<td>84</td>
<td>T</td>
<td>116</td>
<td>t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>NAK</td>
<td>53</td>
<td>5</td>
<td>85</td>
<td>U</td>
<td>117</td>
<td>u</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>SYN</td>
<td>54</td>
<td>6</td>
<td>86</td>
<td>V</td>
<td>118</td>
<td>v</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>ETB</td>
<td>55</td>
<td>7</td>
<td>87</td>
<td>W</td>
<td>119</td>
<td>w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>CAN</td>
<td>56</td>
<td>8</td>
<td>88</td>
<td>X</td>
<td>120</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>EM</td>
<td>57</td>
<td>9</td>
<td>89</td>
<td>Y</td>
<td>121</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>SUB</td>
<td>58</td>
<td>:</td>
<td>90</td>
<td>Z</td>
<td>122</td>
<td>z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>ESC</td>
<td>59</td>
<td>;</td>
<td>91</td>
<td>]</td>
<td>123</td>
<td>{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>FS</td>
<td>60</td>
<td>&lt;</td>
<td>92</td>
<td>\</td>
<td>124</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>GS</td>
<td>61</td>
<td>=</td>
<td>93</td>
<td>}</td>
<td>125</td>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>RS</td>
<td>62</td>
<td>&gt;</td>
<td>94</td>
<td>^</td>
<td>126</td>
<td>~</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>US</td>
<td>63</td>
<td>?</td>
<td>95</td>
<td>_</td>
<td>127</td>
<td>DEL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
String

- Represents a number of letters together
  - Not a native C++ type, must be included using
    ```
    #include <string>
    ```

- Has a number of built-in functions

- Example
  ```
  string myName = "Mordecai";
  ```
Boolean

• Represents a logic state
  – Uses the c++ keyword bool
  – Can hold 2 values: true or false

• Example c++ declaration
  `bool isItSnowing = false;`
What type would you use for?

- Age
- Weight
- Number of people in a room
- Pi
- The middle initial in a person's name
- Grade
- Letter grade
- How many classes you've taken
What type would you use for?

- Year of Birth
- Accelleration due to gravity in m/s^2.
- Number of atoms in a molecule
- Pay rate
- Number of seconds since Jan 1, 1970
- Speed of light
Constants

- When you know a variable is never going to change, use the const keyword at initialization

- 
  const int DAYS_IN_APRIL = 30;
  
- const int INCHES_IN_A_FOOT = 12;
  
- const float pi = 3.14159265358979;
Math Operations

- Operations are defined based on the data types that they work on.

- Ints: 5 main operations
  \[=, +, -, *, /, \%, \+, -, \+=, -=, *=, /=\]

- Char: Same operations as int, may be cast to an int first.
More Math Operations

- Float:
  =, +, -, *, /, ++, --

- String
  =, +
Even More Math Operations

• Booleans
  
  &&, ||, !

• If different types are on other sides of the sign, a conversion will be made to the more flexible type.
Integer Division

• Two types of division
  – Int division – get a whole number (round down)
  – Modulus division – get the remainder
Getting info from the user

#include <iostream>
using namespace std;
int main ( )
{
    int temporaryInteger;
    cin >> temporaryInteger;
    cout << temporaryInteger;
}

• We can finally do useful things!
More ways to get input

- Get a single character:
  ```
  char initial;
  cin.get(initial);
  ```

- Get a whole line including spaces
  ```
  string aSentance;
  getline ( cin, aSentance );
  ```
Doing useful things

Write a program that does something useful...

Write a program which can take the average of any 3 numbers.

Write a program which calculates the volume of a sphere of any radius, $r$. 
Volume of a sphere

Some useful information:

The volume of a sphere is given by:

\((4 / 3) \times \pi \times r \times r \times r\).

\(\pi =\) approximately 3.14169265
Volume of a sphere

Let's start with the standard stuff...

```
#include <iostream>
using namespace std;
int main () {

    cout << “The volume of the sphere is”;
}
```
Volume of a sphere

// We could hard code in a value, eg. 3
#include <iostream>
using namespace std;
int main () {
    cout << "The volume of the sphere is";
    cout << ( 4 / 3 ) * 3.141592 * 3 * 3 * 3;  
}
Volume of a sphere

// Or we can add flexibility with user input
#include <iostream>
using namespace std;
int main () {
    int radius;
    cout << " Enter the radius of the sphere:";  
    cin >> radius;
    cout << "The volume of the sphere is:";
    cout << ( 4 / 3 ) * 3.141592 * radius * radius * radius;
    return 0;
}
But wait, there's a bug in this code...

// Or we can add flexibility with user input
#include <iostream>
using namespace std;

int main () {
    int radius;
    cout << " Enter the radius of the sphere:";
    cin >> radius;
    cout << "The volume of the sphere is:";
    cout << ( 4 / 3 ) * 3.141592 * radius * radius * radius;
    return 0;
}
Find the number of weeks and days

- Write a simple program which finds the numbers of weeks and leftover days for a given integer input.

- Eg. Find the number of weeks and days in: 39
  - 5 Weeks, 4 days.
Calculate the Average

- Find the average of 5 numbers
- The user will enter the 5 numbers.
- Safe assumptions?
Take a look at the following code snippet:

```cpp
int x, y, z, a;
x = 10;
y = 3;
z = 50;
a = (x + y + z) / 3.0
cout << a;
```
The Art of Naming

Take a look at the following code snippet:

```cpp
int firstValue, secondValue, thirdValue, average;
firstValue = 10;
secondValue = 3;
thirdValue = 50;
average = (firstValue + secondValue + thirdValue) / 3.0;
cout << average;
```
TIWIKWITCS110: The Art of Naming

Take a look at the following code snippet:

```cpp
int x = 3;
float y = 3.14;
return (4.0 / 3.0) * y * x * x * x * x;
```

What does this code do?
The Art of Naming

Take a look at the following code snippet:

```c
int r = 3;
float pi = 3.14;
return (4.0 / 3.0) * pi * r * r * r;
```

What does this code do?
Naming Conventions

- Whenever you think of a variable, you need to be mindful of the type
  - A trick to help you think this way is to include a “code letter” in your names representing type.
  - There are a few standards for this:
    - One letter style:
      - int nMyAge;
      - float fSomething;
    - Three letter style:
      - int intDaysUntilXMas;
      - char chrMiddleInitial;
Why follow naming conventions?

- A good name avoids needing as many comments
- Makes it easier for you to read your code
- Makes it easier for others to read your code
- Makes it easier to debug
- Indicates you're thinking about the problem
- Operations are defined by data types
- Tend to get better marks