Data Types

Section 2.7 – 2.12
Today

- Last we covered
  - `main` function
  - `cout` object
  - How data that is used by a program can be declared and stored

- Today we will
  - Investigate the various types of data that C++ can handle
Declaration Statements

• Variable declarations

```c
double area;
double circ;
double perimeter, volume;
```

• Constant declaration

```c
const double PI = 3.14;
const double RADIUS = 5.4;
```
Identifiers

- Programmer-defined names that represent some element of a program

- C++ limits on variable names:
  1. Identifiers must begin with a letter or an underscore
  2. Identifiers must consist of letters, numbers and underscore, nothing else
  3. Identifiers cannot be a keyword
Identifiers

• Identifiers are case sensitive

```c
int totalCost;
int TotalCost;
```

• Use meaningful variable names

```c
width
w
```
Identifiers

Q 4.1 Which of the following declarations are invalid and why?

a. `char Letter1;`

b. `char 1letter;`

c. `double inches, kms;`

d. `double inches*num;`

e. `int joe’s;`

f. `Int cent_per_inch;`

g. `double two-dimensional;`

h. `char hello;`

i. `int return;`

j. `size int;`
Data types

• A data type defines:
  ○ how the computer interprets data in memory

• What? What does memory really look like?
  ○ what is a byte? a bit?
Integers

• The main integer data type is int

• ints are finite (why?)

• An int without a sign (+ or - ) is assumed to be positive

• 2,353 is not an int, 2353 is an int

• Operations?
Integer Data Types

• There are six integer data types, each with a different range and a different size
  ○ what does unsigned mean?

• Range of data types is listed on page 44
• see program 2-17 on page 58 for the above output
Variable Ranges

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>short int</td>
<td>2 bytes</td>
<td>–32,768 to 32,767</td>
</tr>
<tr>
<td>int</td>
<td>4 bytes</td>
<td>–2,147,483,648 to 2,147,483,647</td>
</tr>
<tr>
<td>unsigned int</td>
<td>4 bytes</td>
<td>0 to 4,294,967,295</td>
</tr>
<tr>
<td>long int</td>
<td>4 bytes</td>
<td>–2,147,483,648 to 2,147,483,647</td>
</tr>
</tbody>
</table>

- What is the range of an unsigned short?
- What data type should you use for a person’s age?
- What data type should you use for the temperature on the moon?  
  Why?
- What data type should you use for the size of a music file (mp3)?
char

• The **char** data type is used to store single characters (letters, digits, special characters)
  - ASCII

• Character literals are enclosed in **single quotes**

• Examples of character literals are: ‘A’, ‘a’, ‘*’, ‘2’, ‘$’
### ASCII Character Set

- page 1097

<table>
<thead>
<tr>
<th>Decimal Value</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>&lt;space&gt;</td>
</tr>
<tr>
<td>33</td>
<td>!</td>
</tr>
<tr>
<td>65</td>
<td>A</td>
</tr>
<tr>
<td>66</td>
<td>B</td>
</tr>
<tr>
<td>67</td>
<td>C</td>
</tr>
<tr>
<td>97</td>
<td>a</td>
</tr>
<tr>
<td>98</td>
<td>b</td>
</tr>
<tr>
<td>99</td>
<td>c</td>
</tr>
</tbody>
</table>

- [http://asciitable.com](http://asciitable.com)
Example

// page 48, program 2-11
#include <iostream>

using namespace std;

int main()
{
    char letter;

    letter = 65;
    cout << letter << endl;
    letter = 66;
    cout << letter << endl;
    return 0;
}
Program 4.2

// page 49, program 2-12
#include <iostream>

using namespace std;

int main()
{
    char letter;

    letter = 'A';
    cout << letter << endl;
    letter = 'B';
    cout << letter << endl;
    return 0;
}
string Class

• **string** is used to store a list of characters

• To indicate the end of a **string**, a null terminator is used
  ○ why?

• Need to include the preprocessor directive
  ○ `#include <string>`
  ○ why?
Questions

• Q 4.2 How are the character ‘A’ and the string constant “A” stored in memory?

• Q 4.3 Is the escape character \n a character or a string?

• Q 4.4 How do we declare a char variable and assign it a value?
string Questions

• Q 4.5 How do we declare a variable of type string?

• Q 4.6 How do we assign a value to the variable?

• Q 4.7 How do we output a string constant and a string variable? What is output?
Floating-Point Data Types

- \texttt{float}, \texttt{double}, \texttt{long double}
  - positive and negative
  - no unsigned float!

- Scientific Notation

- Examples:
  - 1.0, -2.3, -0.3, 12E5, -1E-2, 1.4e+8

- 2,353.99 is \textbf{not} a \texttt{double}

- 2353.99 is a \texttt{double}
Variable Sizes

• On my machine the sizes are

```
The size of an int is: 4 bytes.
The size of a short int is: 2 bytes.
The size of a long int is: 4 bytes.
The size of a char is: 1 bytes.
The size of a float is: 4 bytes.
The size of a double is: 8 bytes.
Press any key to continue.
```
Variable Size Program

// page 58, program 2-17
#include <iostream>

using namespace std;

int main()
{
    cout << "The size of an int is:\t\t" << sizeof(int) << " bytes.\n";
    cout << "The size of a short int is:\t" << sizeof(short) << " bytes.\n";
    cout << "The size of a long int is:\t" << sizeof(long) << " bytes.\n";
    cout << "The size of a char is:\t\t" << sizeof(char) << " bytes.\n";
    cout << "The size of a float is:\t\t" << sizeof(float) << " bytes.\n";
    cout << "The size of a double is:\t" << sizeof(double) << " bytes.\n";

    return 0;
}

## Variable Ranges

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>4 bytes</td>
<td>–2,147,483,648 to 2,147,483,647</td>
</tr>
<tr>
<td>short int</td>
<td>2 bytes</td>
<td>–32,768 to 32,767</td>
</tr>
<tr>
<td>long int</td>
<td>4 bytes</td>
<td>–2,147,483,648 to 2,147,483,647</td>
</tr>
<tr>
<td>unsigned int</td>
<td>4 bytes</td>
<td>0 to 4,294,967,295</td>
</tr>
<tr>
<td>char</td>
<td>1 byte</td>
<td>256 character values</td>
</tr>
<tr>
<td>float</td>
<td>4 bytes</td>
<td>±3.4e–38 to ± 3.4e38</td>
</tr>
<tr>
<td>double</td>
<td>8 bytes</td>
<td>±1.7e–308 to ± 1.7e308</td>
</tr>
</tbody>
</table>
How to Choose a Numeric Data Type

• Ask yourself the following questions
  ◦ What are the largest and smallest numbers that may be stored?
  ◦ How much memory does the variable use?
  ◦ Is the variable signed (positive and negative)?
  ◦ How many decimal places of precision does the variable need?
Problem

• What variables will you need for the following program?

• page 71, #4.

• Write a program that computes the tax and tip on a restaurant bill. The user will enter the original bill and the tax rate. Assume a 15% tip. Display the tax amount, tip amount, and total bill on the screen.
Examples

- Remember, the format for declaring variables is:
  - `data-type identifier;`

- You can declare variables of the different data types as follows
  - `int num1;`
  - `double num2;`
  - `char letter;`
Summary

• In today’s lecture we covered
  ○ Identifiers
  ○ Data types
  ○ How data that is used by a program can be declared and stored

• We have covered sections 2.7 – 2.12 of your textbook