| Week 1: Introduction to C++ Gaddis: Chapter 2 (excluding 2.1, 2.11, 2.14) CS 1428 Fall 2014 Jill Seaman | Literals A literal represents a constant value used in a program statement. Numbers: 0, 34, 3.14159, -1.8e12, etc. Characters: 'A', 'z', '!', '5', etc. Strings (sequence of characters): "Hello", "This is a string" "100 years", "100", "Y", etc. NOTE: These are all different: 5, '5', "5" |
|--|--|
| Special characters Newline: '\n' Double quote: '\"'. These can occur in strings: "Hello\nthere" "she said \"boo\" very quietly" See textbook for more It's a backslash (), not a slash () | 2.2 The cout Object cout: short for "console output" a stream object: represents the contents of the screen <<: the stream insertion operator use it to send data to cout (to be output to the screen) cout << "This is an example." when this instruction is executed, the console (screen) looks like this: |

The endl manipulator more examples • end1: short for "end line" cout << "Hello " << "there!";</pre> send it to cout when you want to start a new line of output. Hello there! cout << "Hello " << endl << "there!";</pre> cout << "Hello ";</pre> or you can use the newline character: \n cout << "there!";</pre> cout << "Hello \nthere!";</pre> Hello there! • Either way the output to the screen is: cout << "The best selling book on Amazon\n is \"The Help\""; Hello there! The best selling book on Amazon is "The Help" 5 2 5 Identifiers 2.4 Variables (and Literals) An identifier is a name for some program Variable: named location in main memory element (like a variable). Has a name and a datatype • Rules: <datatype> <identifier> May not be a keyword (see Table 2.4 in the book) • The data type indicates the kind of data it can contain. First character must be a letter or underscore A variable must be defined before it can be Following characters must be letters, numbers or used!! underscores. • Examples: Identifiers are case-sensitive: int someNumber; myVariable is not the same as MyVariable > char firstLetter; 7 8

2.12 Variable Assignments and Initialization

• An **assignment statement** uses the = operator to store a value in an already defined variable.

```
someNumber = 12;
```

- When this statement is executed, the computer stores the value 12 in memory, in the location named "someNumber".
- The variable receiving the value must be on the left side of the = (the following does NOT work):

▶ 12 = someNumber; //This is an ERROR

Variable Initialization

 To initialize a variable means to assign it a value when it is defined:

```
int length = 12;
```

• You can define and initialize multiple variables at once (and change them later) :

```
int length = 12, width = 5, area;
area = 35;
length = 10;
area =40;
```

Example program using a variable

output screen: The value of the number is 100

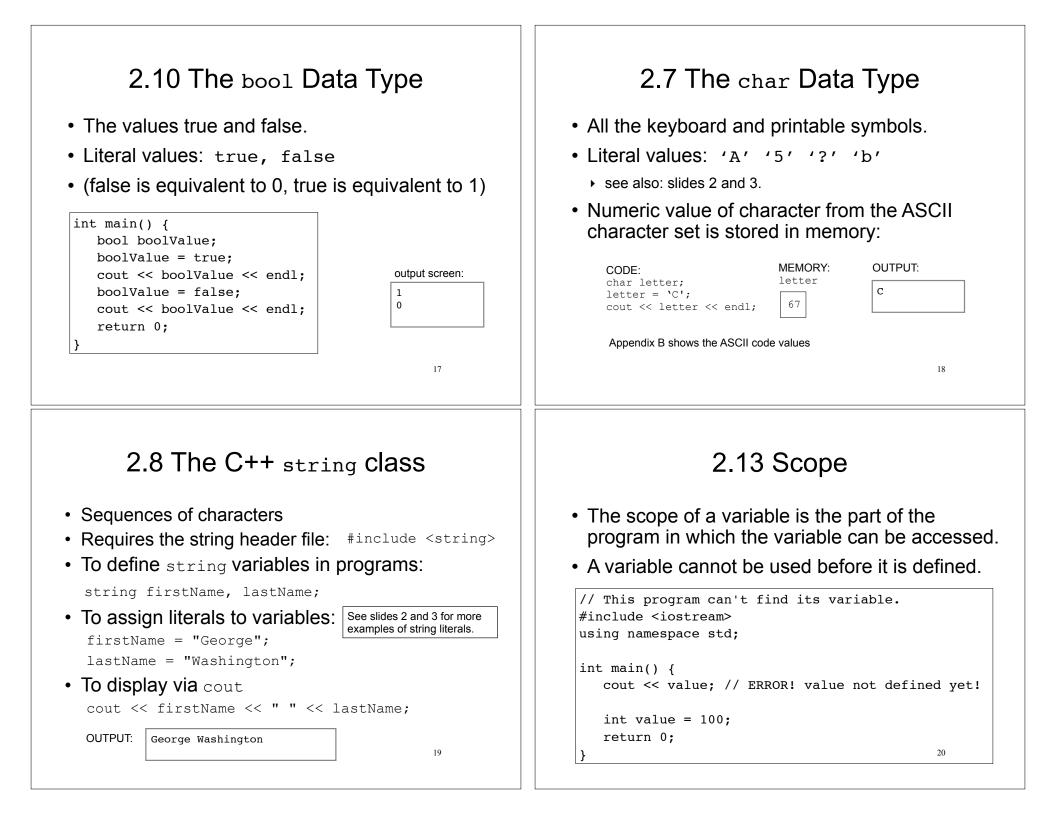
Data Types

- Variables are classified according to their data type.
- The data type determines the kind of information that may be stored in the variable.
- A data type is a set of values.
- Generally two main (types of) data types:
 - Numeric
 - Character-based

9

10

C++ Data Types 2.6 Integer Data Types • int, short, long • Whole numbers such as 12, 7, and -99 whole numbers (integers) Typical ranges (may vary on different systems): • float, double Range of values: Data Type: real numbers (with fractional amounts, decimal points) -32,768 to 32,767 short unsigned short 0 to 65.535 • bool -2,147,483,648 to 2,147,483,647 int unsigned int 0 to 4,294,967,295 lona -2,147,483,648 to 2,147,483,647 Iogical values: true and false 0 to 4.294.967.295 unsigned long • char Example variable definitions: a single character short dayOfWeek; • string unsigned long distance; int xCoordinate: any text, a sequence of characters 13 14 Example program using floating-2.9 Floating-Point Data Types point data types // This program uses floating point data types. • Real numbers such as 12.45, and -3.8 #include <iostream> Typical ranges (may vary on different systems): using namespace std; int main() { Data Type: Range of values: float distance; +/-3.4e +/-38 (~7 digits of precision) float double +/- 1.7e +/- 308 (~15 digits of precision) double mass; long double +/- 1.7e +/- 308 (~15 digits of precision) distance = 1.495979E11; mass = 1.989E30; Floating-point literals can be represented in cout << "The Sun is " << distance << " meters away.\n";</pre> cout << "The Sun\'s mass is " << mass << " kilograms.\n";</pre> - Fixed point (decimal) notation: return 0; 31.4159 0.0000625 - E (scientific) notation: The Sun is 1.49598e+11 meters away. output screen: The Sun's mass is 1.989e+30 kilograms. 3.14159E1 6.25e-5 15 16



2.15 Comments

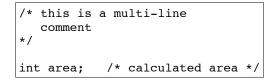
- Used to document parts of the program
- Intended for humans reading the source code of the program:
 - Indicate the purpose of the program
 - Describe the use of variables
 - Explain complex sections of code
- Are ignored by the compiler

Single and Mult-Line Comments

• Single-Line comments begin with // through to the end of line:

```
int length = 12; // length in inches
int width = 15; // width in inches
int area; // calculated area
// calculate rectangle area
area = length * width;
```

• Multi-Line comments begin with /*, end with */



22

2.16 Named Constants

- <u>Named constant</u> : variable whose value cannot be changed during program execution
- Used for representing constant values with descriptive names:

const double TAX_RATE = 0.0675; const int NUM STATES = 50;

Note: initialization required.

 Often named in uppercase letters (see style guidelines)

2.17 Programming Style

- The visual organization of the source code
- Includes the use of spaces, tabs, and blank lines
- Includes naming of variables, constants.
- · Includes where to use comments.
- Purpose: improve the readability of the source code

21

Programming Style

Common elements to improve readability:

- Braces { } aligned vertically
- Indentation of statements within a set of braces
- Blank lines between declaration and other statements
- Long statements wrapped over multiple lines with aligned operators

See the Style Guidelines on the class website. You must follow these in your programming assignments.

25