# Ch 2: Introduction to C++ 

 Part 2CS 1428<br>Fall 2011

Jill Seaman
Lecture 3

## 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


## Data Types

```
int, short, long
    whole numbers
float, double
    real numbers
bool
    logical values: true, false
char
    single character
string
    any text
```


## Integer Data Types

```
int, short, long
```

May be signed (default) or unsigned (signed is a modifier)
Typical sizes and ranges (may vary depending on the system)

| short | 2 bytes | $-32,768$ to 32,767 |
| :--- | :--- | :--- |
| unsigned short | 2 bytes | 0 to 65,535 |
| int | 4 bytes | $-2,147,483,648$ to $2,147,483,647$ |
| unsigned int | 4 bytes | 0 to $4,294,967,295$ |
| long | 4 bytes | $-2,147,483,648$ to $2,147,483,647$ |
| unsigned long | 4 bytes | 0 to $4,294,967,295$ |

Literals (int by default): $\begin{array}{llllllll}0 & 5 & 100 & -3 & 2147000001 & -2434555 & 1\end{array}$
Example variable definitions:
short dayOfWeek;
unsigned long distance;
int xCoordinate;

## Floating-Point Data Types

float, double, long double

Typical sizes and ranges (may vary depending on the system):

| float | 4 bytes | $+/-3.4 \mathrm{e}+/-38(\sim 7$ digits $)$ |
| :--- | :--- | :--- |
| double | 8 bytes | $+/-1.7 \mathrm{e}+/-308(\sim 15$ digits $)$ |
| long double | 8 bytes | $+/-1.7 \mathrm{e}+/-308(\sim 15$ digits $)$ |

Literals (default type is double):

| 31.415E5 | // equivalent to 3141500.0 |
| :--- | :--- |
| -31.415 e 5 | // equivalent to -3141500.0 |
| $3.1 \mathrm{e}-4$ | // equivalent to 0.00031 |
| 31000.0 | // NO, this is an int literal |
| 31000 | // converts to float |
| 31000.0f | error, must be a floating-point literal |
| 31000f |  |

## Floating-Point Data Types

| +3.4 e 38 | is equal to | $34000 \ldots 000$ | (with 37 zeros) |
| :--- | :--- | :--- | :--- | :--- |
| -3.4 e 38 | is equal to | $-34000 \ldots 000$ | (with 37 zeros) |
| $+3.4 \mathrm{e}-38$ | is equal to | $.000 \ldots 034$ | (with 37 zeros) |
| $-3.4 \mathrm{e}-38$ | is equal to | $-.000 \ldots . .034$ | (with 37 zeros) |

maximum vs minimum, biggest vs smallest (absolute value)
"~7 digits"
number of digits in the mantissa:
$-1.234567 \mathrm{e}-25$ is equal to $.000 \ldots 01234567$ (with 24 zeros)
This number: 0.1234567890123456789
will get rounded to: 0.1234568
which is the same as: 0.1234568 e 0
If it is stored as a double, it will get rounded to:
0.123456789012346
because it can store about 15 digits in the mantissa.

## Floating-Point Data Types

```
float distance, time;
double mass;
distance = 1.495979E11; // how far away the sun is (in meters)
mass=1.989E30; // how much the sun weighs (in kilograms)
time = 12.816; }\quad// hours of daylight in San Marcos today, 8/3
Converting between floating-points and integers:
int i;
float f;
\(\mathrm{f}=8.9\);
\(i=8.9 ; \quad / /\) stores 8 in \(i\) (truncates, does not round)
\(\mathrm{i}=8\);
\(\mathrm{f}=8\); \(\quad / /\) stores 8.0 in f
\(\mathrm{f}=7.9\);
\(\mathrm{i}=\mathrm{f} ; \quad / /\) stores 7 in i
```


## The bool Data Type

- bool
- Literals: true false

```
bool boolValue;
boolValue = true;
cout << boolValue << endl;
boolValue = false;
cout << boolValue << endl;
Output:
1
0
```

- bool is a numeric type:
- true is 1 and false is 0


## The char Data Type

- char
- Literals: 'A' '3' '!' 'ln' 'n'

```
char letter;
```

letter = 'A';
cout << letter << endl;
letter = '!';
cout << letter << endl;
Output:
A
!

## The char Data Type

- char is really a numeric type also!
- Note: 65 is the ASCII code for ' $\mathrm{A}^{\prime}$

```
char letter;
letter = 65;
cout << letter << endl;
letter = 66;
cout << letter << endl;
Output:
A
B
```


## The string Data Type

- We will not declare variables that contain strings yet.
- A string is a sequence of characters.
- Literals: "Hello again" "OverlnThere" " $\gamma$ "
- A string is stored sequentially in memory, with the null character ('IO') at the end.
- The null character is not displayed.

