

## Data Types

Section 2.7 – 2.12

# Today

---

- Last time 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

```
double area;
```

```
double circle;
```

```
double perimeter, volume;
```

- Constant declaration

```
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*

**page 42**

# Identifiers

---

- Identifiers are case sensitive

```
int totalCost;
```

```
int TotalCost;
```

- Use meaningful variable names

```
width
```

```
w
```

# Identifiers

---

- Q 4.1 Which of the following declarations are invalid and why?
  - `char Letter1;`
  - `char 1letter;`
  - `double inches, kms;`
  - `double inches*num;`
  - `int joe's;`
  - `Int cent_per_inch;`
  - `double two-dimensional;`
  - `char hello;`
  - `int return;`
  - ~~`double inches; kms;`~~

# 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

---

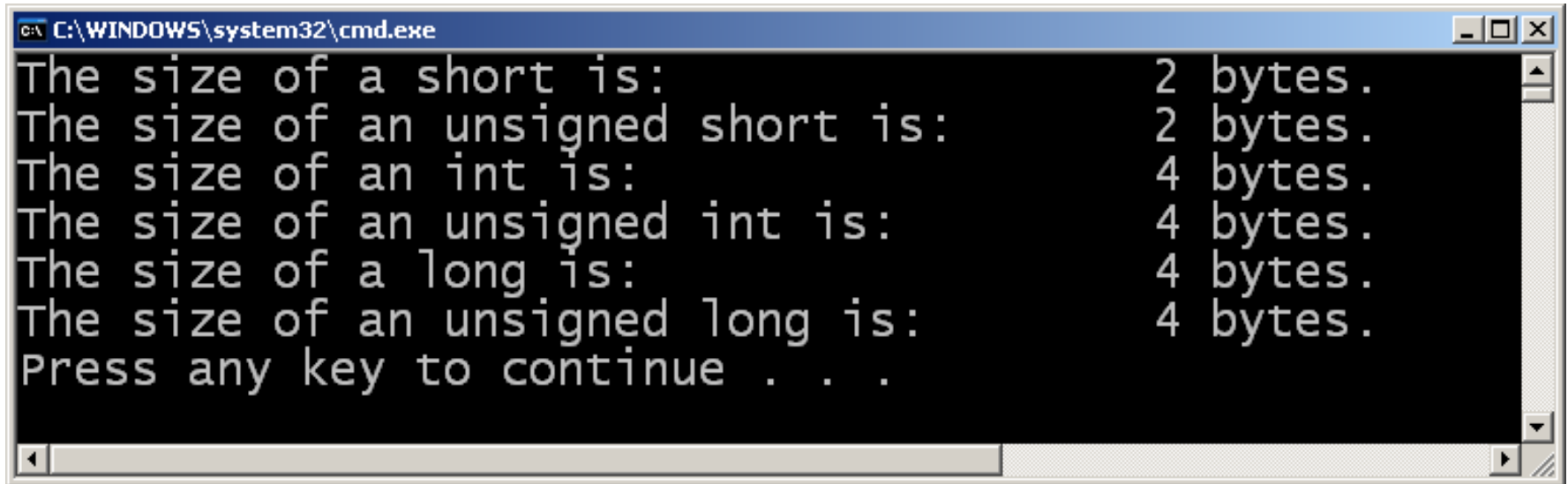
- The main integer data type is `int`
- `ints` are `finite` (why?)
- An `int` without a sign (+ or - ) is assumed to be positive
- 2,353 is not an `int`, 2353 is an `int`
- Operations?



# Integer Data Types

---

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



```
C:\WINDOWS\system32\cmd.exe
The size of a short is:          2 bytes.
The size of an unsigned short is: 2 bytes.
The size of an int is:          4 bytes.
The size of an unsigned int is:  4 bytes.
The size of a long is:          4 bytes.
The size of an unsigned long is: 4 bytes.
Press any key to continue . . .
```

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

# Variable Ranges

---

Type	Size	Values
<code>short int</code>	2 bytes	-32,768 to 32,767
<code>int</code>	4 bytes	-2,147,483,648 to 2,147,483,647
<code>unsigned int</code>	4 bytes	0 to 4,294,967,295
<code>long int</code>	4 bytes	-2,147,483,648 to 2,147,483,647

- 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 population of the earth?
- What data type should you use for the number of students at Pacific University?

**Why?**

# 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

Decimal Value	Character
32	<space>
33	!
65	A
66	B
67	C
97	a
98	b
99	c

- <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

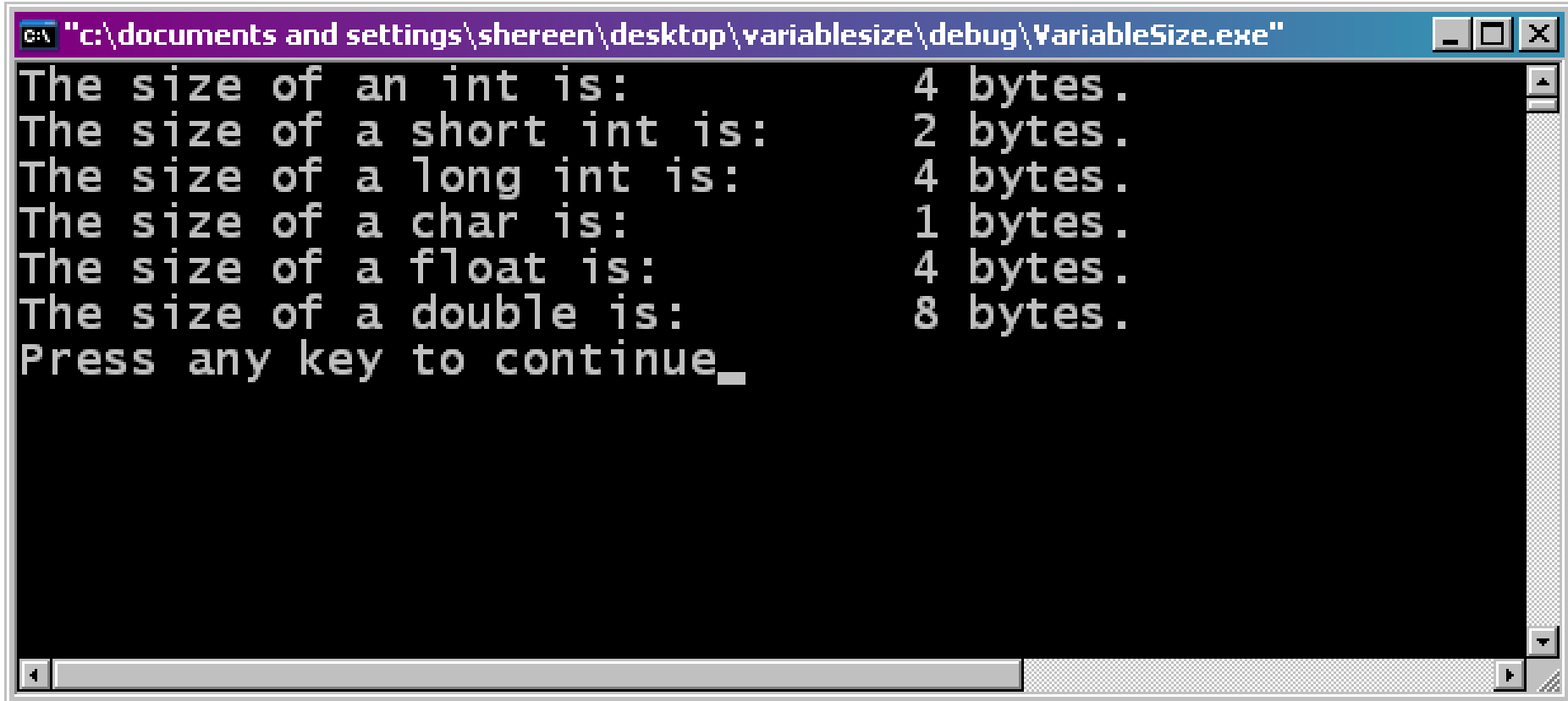
---

- **float, double, 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 **not** a **double**
- 2353.99 is a **double**

# Variable Sizes

---

- On my machine the sizes are

A screenshot of a Windows command prompt window. The title bar reads "c:\documents and settings\shereen\desktop\variablesizedebug\VariableSize.exe". The window contains the following text:

```
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

---

Type	Size	Values
<code>int</code>	4 bytes	-2,147,483,648 to 2,147,483,647
<code>short int</code>	2 bytes	-32,768 to 32,767
<code>long int</code>	4 bytes	-2,147,483,648 to 2,147,483,647
<code>unsigned int</code>	4 bytes	0 to 4,294,967,295
<code>char</code>	1 byte	256 character values
<code>float</code>	4 bytes	$\pm 3.4e^{-38}$ to $\pm 3.4e^{38}$
<code>double</code>	8 bytes	$\pm 1.7e^{-308}$ to $\pm 1.7e^{308}$

# 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