



Lecture 4:
**Introduction to
C++ Programming**

Ioan Raicu

Department of Electrical Engineering & Computer Science
Northwestern University

EECS 211
Fundamentals of Computer Programming II
April 2nd, 2010

2.2 First Program in C++: Printing a Line of Text

- Simple program that prints a line of text (Fig. 2.1).

2.2 First Program in C++: Printing a Line of Text

```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome to C++!\n"; // display message
9
10    return 0; // indicate that program ended successfully
11 } // end function main
```

Welcome to C++!

Fig. 2.1 | Text-printing program.

2.2 First Program in C++: Printing a Line of Text (cont.)

- `//` indicates that the remainder of each line is a **comment**.
 - You insert comments to document your programs and to help other people read and understand them.
 - Comments are ignored by the C++ compiler and do not cause any machine-language object code to be generated.
- A comment beginning with `//` is called a **single-line comment** because it terminates at the end of the current line.
- You also may use C's style in which a comment—possibly containing many lines—begins with `/*` and ends with `*/`.

2.2 First Program in C++: Printing a Line of Text



Good Programming Practice 2.1

Every program should begin with a comment that describes the purpose of the program.

2.2 First Program in C++: Printing a Line of Text (cont.)

- A **preprocessor directive** is a message to the C++ preprocessor.
- Lines that begin with **#** are processed by the preprocessor before the program is compiled.
- **#include <iostream>** notifies the preprocessor to include in the program the contents of the **input/output stream header file <iostream>**.
 - Must be included for any program that outputs data to the screen or inputs data from the keyboard using C++-style stream input/output.

2.2 First Program in C++: Printing a Line of Text



Common Programming Error 2.1

Forgetting to include the `<iostream>` header file in a program that inputs data from the keyboard or outputs data to the screen causes the compiler to issue an error message, because the compiler cannot recognize references to the stream components (e.g., `cout`).

2.2 First Program in C++: Printing a Line of Text (cont.)

- You use blank lines, space characters and tab characters (i.e., “tabs”) to make programs easier to read.
 - Together, these characters are known as **white space**.
 - White-space characters are normally ignored by the compiler.

2.2 First Program in C++: Printing a Line of Text



Good Programming Practice 2.2

Use blank lines, space characters and tabs to enhance program readability.

2.2 First Program in C++: Printing a Line of Text (cont.)

- `main` is a part of every C++ program.
- The parentheses after `main` indicate that `main` is a program building block called a `function`.
- C++ programs typically consist of one or more functions and classes.
- Exactly one function in every program must be named `main`.
- C++ programs begin executing at function `main`, even if `main` is not the first function in the program.
- The keyword `int` to the left of `main` indicates that `main` “returns” an integer (whole number) value.
 - A `keyword` is a word in code that is reserved by C++ for a specific use.
 - For now, simply include the keyword `int` to the left of `main` in each of your programs.

2.2 First Program in C++: Printing a Line of Text (cont.)

- A **left brace**, {, must begin the **body** of every function.
- A corresponding **right brace**, }, must end each function's body.
- A statement instructs the computer to **perform an action**.
- A string is sometimes called a **character string** or a **string literal**.
- We refer to characters between double quotation marks simply as **strings**.
 - White-space characters in strings are not ignored by the compiler.
- A statement normally ends with a **semicolon** (;), also known as the **statement terminator**.
 - Preprocessor directives (like `#include`) do not end with a semicolon.

2.2 First Program in C++: Printing a Line of Text (cont.)

- When a `cout` statement executes, it sends a stream of characters to the **standard output stream object**—`std::cout`—which is normally “connected” to the screen.
- The `std::` before `cout` is required when we use names that we’ve brought into the program by the preprocessor directive `#include <iostream>`.
 - The notation `std::cout` specifies that we are using a name, in this case `cout`, that belongs to “namespace” `std`.
 - The names `cin` (the standard input stream) and `cerr` (the standard error stream) also belong to namespace `std`.
- The `<<` operator is referred to as the **stream insertion operator**.
 - The value to the operator’s right, the right **operand**, is inserted in the output stream.

2.2 First Program in C++: Printing a Line of Text (cont.)

- The characters `\n` are not printed on the screen.
- The backslash (`\`) is called an **escape character**.
 - It indicates that a “special” character is to be output.
- When a backslash is encountered in a string of characters, the next character is combined with the backslash to form an **escape sequence**.
- The escape sequence `\n` means **newline**.
 - Causes the **cursor** to move to the beginning of the next line on the screen.
- When the **return statement** is used at the end of `main` the value `0` indicates that the program has terminated successfully.
- According to the C++ standard, if program execution reaches the end of `main` without encountering a **return** statement, it’s assumed that the program terminated successfully—exactly as when the last statement in `main` is a **return** statement with the value `0`.

2.2 First Program in C++: Printing a Line of Text

Escape sequence	Description
<code>\n</code>	Newline. Position the screen cursor to the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the screen cursor to the next tab stop.
<code>\r</code>	Carriage return. Position the screen cursor to the beginning of the current line; do not advance to the next line.
<code>\a</code>	Alert. Sound the system bell.
<code>\\</code>	Backslash. Used to print a backslash character.
<code>\'</code>	Single quote. Use to print a single quote character.
<code>\"</code>	Double quote. Used to print a double quote character.

Fig. 2.2 | Escape sequences.

2.2 First Program in C++: Printing a Line of Text



Common Programming Error 2.2

Omitting the semicolon at the end of a C++ statement is a syntax error. (Again, preprocessor directives do not end in a semicolon.) The **syntax** of a programming language specifies the rules for creating proper programs in that language. A **syntax error** occurs when the compiler encounters code that violates C++'s language rules (i.e., its syntax). The compiler normally issues an error message to help you locate and fix the incorrect code. Syntax errors are also called **compiler errors**, **compile-time errors** or **compilation errors**, because the compiler detects them during the compilation phase. You cannot execute your program until you correct all the syntax errors in it. As you'll see, some compilation errors are not syntax errors.

2.2 First Program in C++: Printing a Line of Text



Good Programming Practice 2.3

Indent the entire body of each function one level within the braces that delimit the body of the function. This makes a program's functional structure stand out and makes the program easier to read.

2.2 First Program in C++: Printing a Line of Text



Good Programming Practice 2.4

Set a convention for the size of indent you prefer, then apply it uniformly. The tab key may be used to create indents, but tab stops may vary. We recommend using either 1/4-inch tab stops or (preferably) three spaces to form a level of indent.

2.3 Modifying Our First C++ Program

- `welcome to C++!` can be printed several ways.

2.3 Modifying Our First C++ Program

```
1 // Fig. 2.3: fig02_03.cpp
2 // Printing a line of text with multiple statements.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome ";
9     std::cout << "to C++!\n";
10 } // end function main
```

Welcome to C++!

Fig. 2.3 | Printing a line of text with multiple statements.

2.3 Modifying Our First C++ Program (cont.)

- A single statement can print multiple lines by using newline characters.
- Each time the `\n` (newline) escape sequence is encountered in the output stream, the screen cursor is positioned to the beginning of the next line.
- To get a blank line in your output, place two newline characters back to back.

2.3 Modifying Our First C++ Program

```
1 // Fig. 2.4: fig02_04.cpp
2 // Printing multiple lines of text with a single statement.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome\nto\n\nC++!\n";
9 } // end function main
```

```
Welcome
to

C++!
```

Fig. 2.4 | Printing multiple lines of text with a single statement.

2.4 Another C++ Program: Adding Integers

- The input stream object `std::cin` and the stream extraction operator-, `>>`, can be used obtain data from the user at the keyboard.

2.4 Another C++ Program: Adding Integers (cont.)

```
1 // Fig. 2.5: fig02_05.cpp
2 // Addition program that displays the sum of two integers.
3 #include <iostream> // allows program to perform input and output
4
5 // function main begins program execution
6 int main()
7 {
8     // variable declarations
9     int number1; // first integer to add
10    int number2; // second integer to add
11    int sum; // sum of number1 and number2
12
13    std::cout << "Enter first integer: "; // prompt user for data
14    std::cin >> number1; // read first integer from user into number1
15
16    std::cout << "Enter second integer: "; // prompt user for data
17    std::cin >> number2; // read second integer from user into number2
18
19    sum = number1 + number2; // add the numbers; store result in sum
20
21    std::cout << "Sum is " << sum << std::endl; // display sum; end line
22 }
```

Fig. 2.5 | Addition program that displays the sum of two integers entered at the keyboard. (Part 1 of 2.)

2.4 Another C++ Program: Adding Integers (cont.)

```
Enter first integer: 45  
Enter second integer: 72  
Sum is 117
```

Fig. 2.5 | Addition program that displays the sum of two integers entered at the keyboard. (Part 2 of 2.)

2.4 Another C++ Program: Adding Integers (cont.)

- **Declarations** introduce identifiers into programs.
- The identifiers `number1`, `number2` and `sum` are the names of **variables**.
- A variable is a location in the computer's memory where a value can be stored for use by a program.
- Variables `number1`, `number2` and `sum` are data of type **int**, meaning that these variables will hold **integer** values, i.e., whole numbers such as 7, -11, 0 and 31914.
- All variables must be declared with a name and a data type before they can be used in a program.
- If more than one name is declared in a declaration (as shown here), the names are separated by commas (,); this is referred to as a **comma-separated list**.

2.4 Another C++ Program: Adding Integers (cont.)

```
Enter first integer: 45  
Enter second integer: 72  
Sum is 117
```

Fig. 2.5 | Addition program that displays the sum of two integers entered at the keyboard. (Part 2 of 2.)

2.4 Another C++ Program: Adding Integers (cont.)



Good Programming Practice 2.5

Place a space after each comma (,) to make programs more readable.

2.4 Another C++ Program: Adding Integers (cont.)

- Data type `double` is for specifying real numbers, and data type `char` for specifying character data.
- Real numbers are numbers with decimal points, such as 3.4, 0.0 and -11.19.
- A `char` variable may hold only a single lowercase letter, a single uppercase letter, a single digit or a single special character (e.g., \$ or *).
- Types such as `int`, `double` and `char` are called **fundamental types**.
- Fundamental-type names are keywords and therefore must appear in all lowercase letters.
- Appendix C contains the complete list of fundamental types.

2.4 Another C++ Program: Adding Integers (cont.)

- A variable name is any valid **identifier** that is not a keyword.
- An identifier is a series of characters consisting of letters, digits and underscores (`_`) that does not begin with a digit.
- C++ is **case sensitive**—uppercase and lowercase letters are different, so `a1` and `A1` are different identifiers.

Questions

