



CS150 Intro to CS I

Fall 2012

Chapter 2

Introduction to C++

- Reading: Chapter 2 (2.1 to 2.3)
- Good Problems to Work: p. 37 [2.3, 2.4]

Parts of a C++ Program

- The C++ Hello World program consists of the following elements:
 - Comments
 - Preprocessor directives
 - Standard namespace
 - main function
 - Declaration statements
 - Executable statements

Program Skeleton

- All programs in the beginning of this course should have the following C++ skeleton

```

//*****
// File name:  filename.cpp
// Author:     Your Name
// Date:       Completion Date
// Class:      CS 150-02
// Assignment: Title of assignment
// Purpose:    Description about what the program does
//*****
#include <iostream>
#include <string>

using namespace std;

int main()
{
    // declaration statements

    // executable statements

    return EXIT_SUCCESS;
}

```

Comments

- Comments are
 - how you explain in English what the different parts of your program do
 - ignored by the compiler
 - very important for you and me
- The editor in Visual Studio will color code your comments. Comments will be green.

Comments

- There are two ways to write comments

// I am a comment

- Anything after `//` to the end of the line will be a comment

/* I am another comment */

- You must start the comment with `/*` and end it with `*/` in this style of comment

Preprocessor Directives

- These need to appear at the beginning of every program that you write
- Includes C++ libraries into your program
- Examples:

```
#include <iostream>
```

```
#include <string>
```

namespace std

`using namespace std;`

- The statement appears in all of our programs
- Programs contain several items with unique names (i.e. variables, functions, ..)
- namespaces are used to organize these names
- The statement (`using namespace std;`) declares that the program will be accessing entities whose names are part of the namespace called `std`

main function

```
int main()  
{  
    // program statements  
    return EXIT_SUCCESS;  
}
```

- main is where program execution begins
- return EXIT_SUCCESS; ends the main function and indicates that the program terminated successfully where EXIT_SUCCESS is a predefined constant that is returned
- Everything within the double braces {} must be indented 2 spaces for correct program style

Program Statements

- There are two types of statements that you can write inside the main (or any other) function
 - Declaration statements
 - Specify the **data** that is needed by the program
 - Executable statements
 - Perform **operations**
- All statements must end with a semicolon;

Program Statements

- Declaration statements

```
string name;
```

- Executable statements

```
cout << "Type your name, then press enter" << endl;  
cin >> name;  
cout << "Hello " << name << "!" << endl;
```

cout object

- cout is an executable statement
- cout is the standard output object
- The monitor is the standard output device
- cout is a stream object and works with streams of data
- The executable statement
`cout << "Hello " << endl;`
places what into the output stream?

cout object

- Output operator (insertion operator): <<
- Standard output (monitor, screen): cout

```
cout << "Hello ";
```

- right operand can be a string literal or variable

cout object

- Program Segment #1

```
cout << "one" << "two" << endl;
```

- Program Segment #2

```
cout << "one" << endl << "two" << endl;
```

- What is the purpose of the endl?
- What goes into the stream?
- What is the output?

cout object

- Separate components with <<

```
int age;  
age = 18;  
cout << "Sara is " << age  
      << "years old" << endl;
```

- Don't break string literals across a line as this causes a compiler error

```
cout << "Sara  
      is " << age << "years old" << endl;
```

Escape Characters

- These are special characters that can be output
 - escape characters are part of a string literal
- They are always preceded by a backslash \
- Examples of escape characters include:
 - `\n`: new line: equivalent to `endl`
 - `\r`: moves the cursor to the beginning of the current line
 - `\t`: moves the cursor to the next tab stop
 - `\\`: displays the backslash
 - `\"`: outputs the double quotes
 - `\a`: outputs a beep!

Examples

- What is the output if a tab is set to two spaces?

```
cout << "A C++ program\n";
```

```
cout << "A \nC++ program";
```

```
cout << "\"A C++ program\"";
```

```
cout << "A\tC++\n\tprogram";
```

Tough Question

- What is the output produced by executing the following C++ statement?

```
cout << "\\\" << endl << "\n";
```