

Your First C++ Program

September 1, 2010

Your First C++ Program

```
//*****  
// File name: hello.cpp  
// Author:    Bob Smith  
// Date:     09/01/2010  
// Purpose:   This program displays a welcome message to  
//           the user  
//*****  
#include <iostream>  
#include <string>  
  
using namespace std;  
  
int main()  
{  
    string name;  
  
    cout << "Type your name, then press enter " << endl;  
    cin >> name;  
    cout << "Hello " << name << "! " << endl;  
  
    return 0;  
}
```

Program Output:

```
Type your name, then press enter  
Doug  
Hello Shereen!
```

Language Elements

- Key Words
 - Have special meaning in C++
 - `using namespace int`
- Programmer-Defined Identifiers
 - Names made up by the programmer
 - `name`
- Operators
 - Perform operations
 - `* =`
- Punctuation
 - Used to mark the beginning and end of the program

Syntax (Grammar)

- Rules that must be followed when constructing a program
- Controls the use of key words, programmer-defined identifiers, operators, and punctuation

Program Components

- The C++ program on the previous slide consists of the following elements:
 - Comments
 - Preprocessor directives
 - Standard namespace
 - `main` function
 - Declaration statements
 - Executable statements

Program Skeleton

- All programs in C++ should have the following skeleton

```
//*****  
// File name: filename.cpp  
// Author:    Your Name  
// Date:     09/01/2010  
// Purpose:   Description about what the program does  
//*****  
#include <iostream>  
#include <string>  
  
using namespace std;  
  
int main()  
{  
    // declaration statements  
  
    // executable statements  
  
    return 0;  
}
```

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. They 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

- `#include <iostream>`
- `#include <string>`
- These need to appear at the beginning of every program that you write
- Includes C++ libraries into your program

Namespace std

- `using namespace std;`
- The statement appears in all our programs

main Function

```
int main()  
{  
    // program statements  
    return 0;  
}
```

- Where the start of your program execution begins
- `return 0;` ends the main function and indicates that the program terminated successfully
- Everything within the double braces `{}` should be indented
 - 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;
```

Executable Statements

cout Object

cout Object

- `cout` object is the standard output object
- The monitor is the standard output device
- `cout` is a stream object and works with streams of data
 - Streams of characters
 - Streams?

cout Object

- Output operator (insertion operator): `<<`
- Standard output (monitor screen): `cout`
- `cout << "Hello out there!";`
 - right operand
 - string literal
 - variable

cout Object

- What is the output?

```
cout << "Type your name, then press enter" << endl;
```

- `endl` will move the cursor to a new line
- Statement must end in a semicolon

cout Object

- Other ways of outputting the same message

```
cout << "Type your name, " << "then press enter" << endl;
```

```
cout << "Type your name, ";
```

```
cout << "then press enter" << endl;
```

- Everything will output to the same line unless you specify otherwise

cout Object

- Separate components with <<

```
int kms;  
kms = 4;  
cout << "Enter the distance in miles" << endl;  
cout << "The distance in kilometers is " << kms << endl;
```

- Don't break string literals across a line

```
cout << "Type your name, then  
press enter" << endl;
```

- Compiler error!

Problem

- What is the output?

```
cout << "My name is: ";  
cout << "Doe, Jane." << endl;  
cout << "I live in ";  
cout << "Ann Arbor, MI ";  
cout << "and my zip code is "  
    << 48109 << ". " << endl;
```

Escape Characters

- These are special characters that can be output
 - 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?

```
cout << "This is a C++ program\n";
```

```
cout << "This is a \nC++ program";
```

```
cout << "\"This is a C++ program\"";
```

```
cout << "This is a\tC++\tprogram";
```

Summary

- Today we covered
 - The basic components of a program
 - Program skeleton
 - cout Object
- Next time
 - Data types
 - Identifiers
- Completed sections 2.1-2.3, 2.14