Your First C++ Program

Your First C++ Program

```
//************************************************************
// File name:  hello.cpp
// Author:     Bob Smith
// Date:       08/29/2011
// Class:      CS 150-02
// Assignment: Assignment 0.1
// Purpose:    This program displays a welcome message to
//             the user after the user enters their name
//************************************************************
#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 EXIT_SUCCESS;
}
```

Program Output:

```
Type your name, then press enter
Shereen
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 { }
  - Used to separate C++ statements ;

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 we just looked at 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

- `#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
- 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 `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
  - 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
  - For example, streams of characters

`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 year;
year = 2011;
cout << "The year is " << year << endl;
```

- Don't break string literals across a line

```
cout << "The year
is " << year << 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 if a tab is set to eight spaces?

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

Tough Question

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

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

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