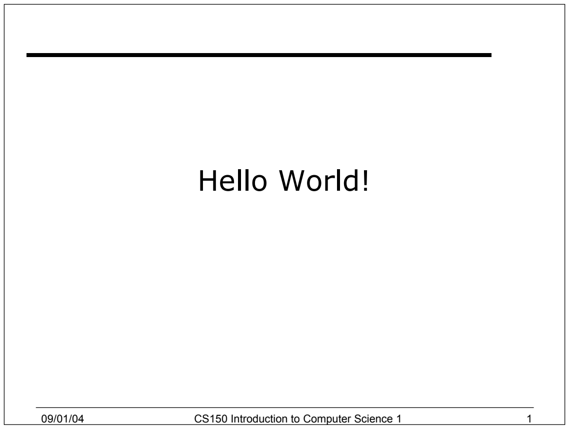
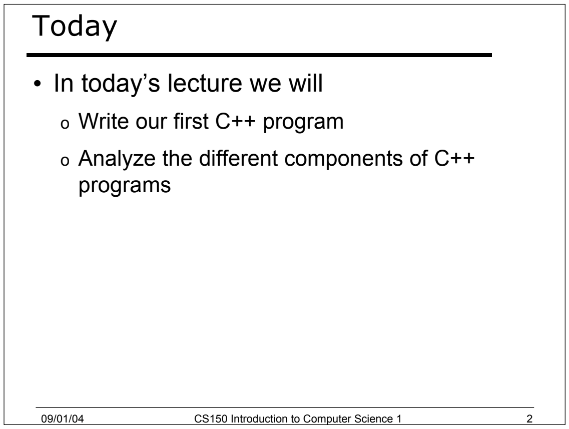
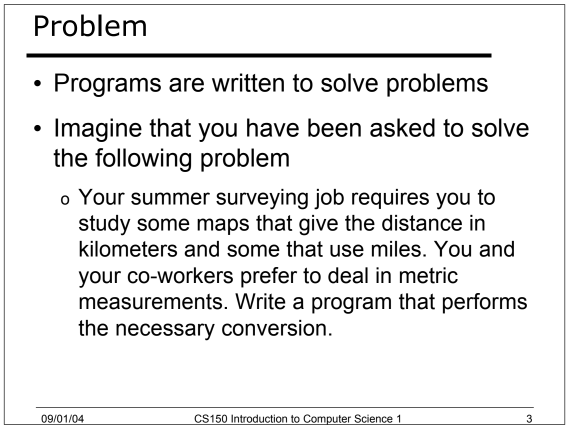


Hello World!



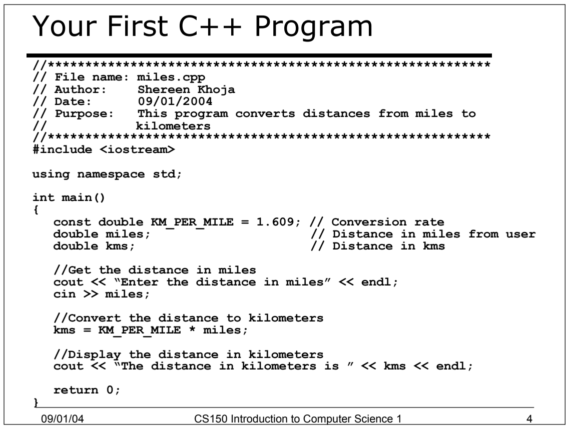
Today

- In today’s lecture we will
  - Write our first C++ program
  - Analyze the different components of C++ programs



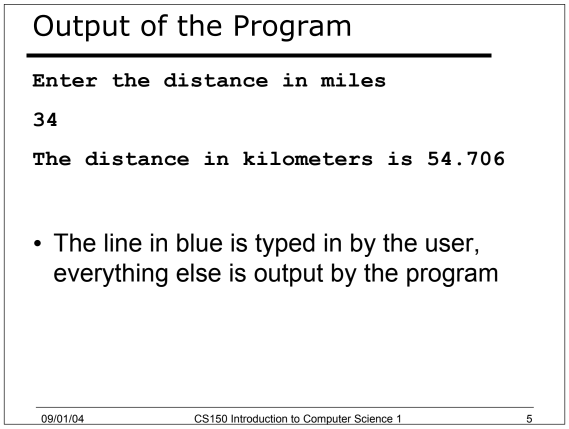
Problem

- Programs are written to solve problems
- Imagine that you have been asked to solve the following problem
  - Your summer surveying job requires you to study some maps that give the distance in kilometers and some that use miles. You and your co-workers prefer to deal in metric measurements. Write a program that performs the necessary conversion.



Your First C++ Program

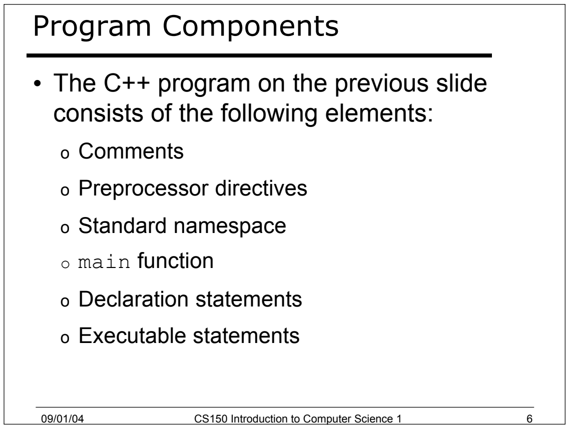
```
//***********************************************************
// File name: miles.cpp
// Author:    Shereen Khoja
// Date:      09/01/2004
// Purpose:   This program converts distances from miles to
//           kilometers
//***********************************************************
#include <iostream>

using namespace std;

int main()
{
   const double KM_PER_MILE = 1.609; // Conversion rate
   double miles;                      // Distance in miles from user
   double kms;                        // Distance in kms

   //Get the distance in miles
   cout << “Enter the distance in miles” << endl;
   cin >> miles;

   //Convert the distance to kilometers
   kms = KM_PER_MILE * miles;

   //Display the distance in kilometers
   cout << “The distance in kilometers is ” << kms << endl;

   return 0;
}
```



Output of the Program

```
Enter the distance in miles

34

The distance in kilometers is 54.706
```

- The line in blue is typed in by the user, everything else is output by the program



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

Comments

- Comments are
  - How you explain in English what the different parts of your program do
  - Ignored by the compiler
  - Very important
- The editor in Visual Studio will colour code your comments. They will be green

Comments

- There are two ways to write comments
  - `// I am a comment`
    - Anything after `//` till 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>`
- `#` signifies preprocessor directive
- Processed before program translation
- `#include` tells the preprocessor to look for libraries
- `<>` signifies part of standard C++ libraries
- We'll see other examples of preprocessor directives later

Preprocessor directives

- `iostream` is the input/output stream library
- It is needed to output data to the screen and read in data from the keyboard
- `#include` takes the contents of the library file and places them in the current program

Namespace std

- `using namespace std;`
- Indicates that we will be using objects (`cout` & `cin`) that are named in a region called `std`
- The statement ends in a semicolon
- The statement appears in all our programs

Namespace std

- You could omit the statement `using namespace std;` from the top of your program
- If you do, then every time you need to use an object from the standard namespace you will need to place `std::` before it
  - `std::cout << “Hello World!”;`

`main` Function

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

- Every program must have a main function
- It is 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

Program Statements

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

Program Statements

- Declaration statements
    - o `const double KM_PER_MILE = 1.609;`
    - o `double miles;`
    - o `double kms;`
- Executable statements
    - o `cout << "Enter the distance in miles" << endl;`
    - o `cin >> miles;`
    - o `kms = KM_PER_MILE * miles;`
    - o `cout << "The distance in kilometers is" << kms << endl;`

Program Skeleton

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

```
//************************************************************
// File name: filename.cpp
// Author:    Your Name
// Date:      09/01/2004
// Purpose:   Description about what the program does
//************************************************************
#include <iostream>

using namespace std;

int main()
{
  // declaration statements

  // executable statements

  return 0;
}
```

Problem

- Write a program that asks the user to enter the radius of a circle and then computes and displays the circle's area
- Write the basic skeleton of this program

Summary

- Today we
    - o Wrote our first C++ program
    - o Introduced the basic components of a C++ program
- To see the program in action you should test it in Visual Studio .NET.
- We covered p. 21 - 26 from your textbook