Opening a Command Shell

In WinXP, go to the Programs Menu, select Accessories and then Command Prompt.

In Win7, go to the All Programs, select Accessories and then Command Prompt.

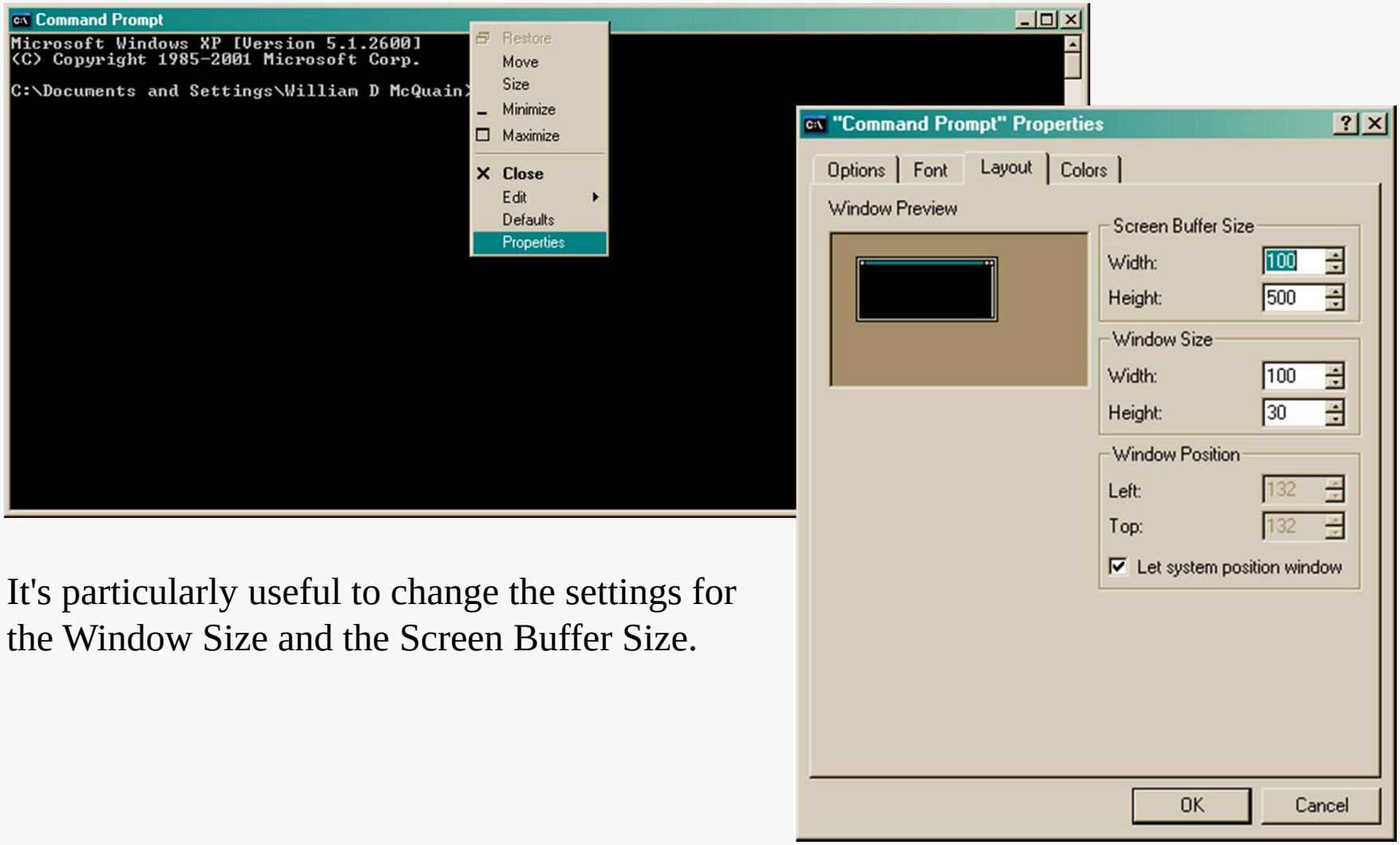
Note you can drag a shortcut off the Programs Menu to your Desktop (copy, not move)… that makes it faster to just open a command shell.

BTW, a command shell window is often referred to as a *console*.

©2010 McQuain

Customizing the Command Shell

Right-click the title bar and select Properties; you can customize many appearance and some operational properties from here…

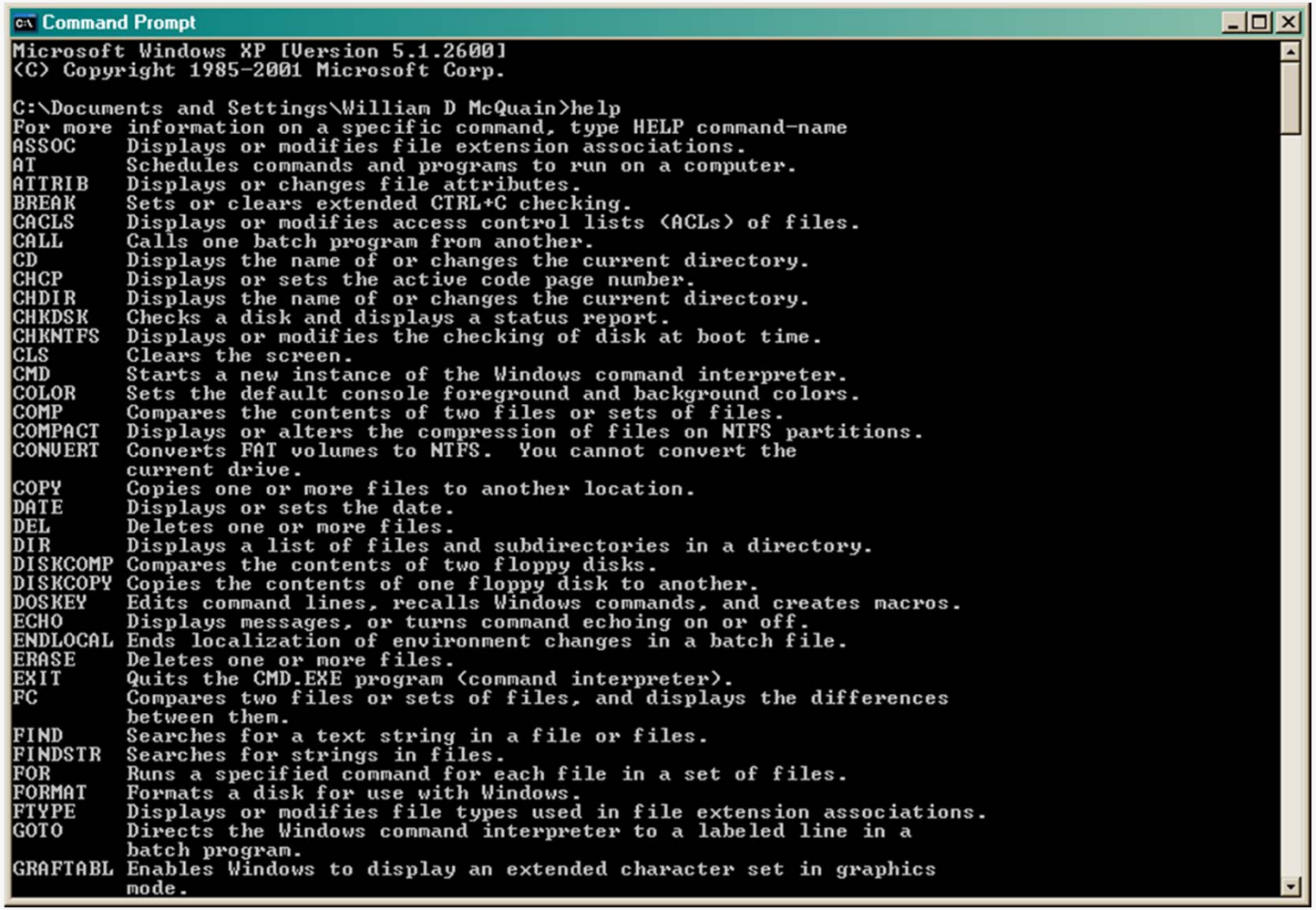


It's particularly useful to change the settings for the Window Size and the Screen Buffer Size.

©2010 McQuain

Common Shell Commands

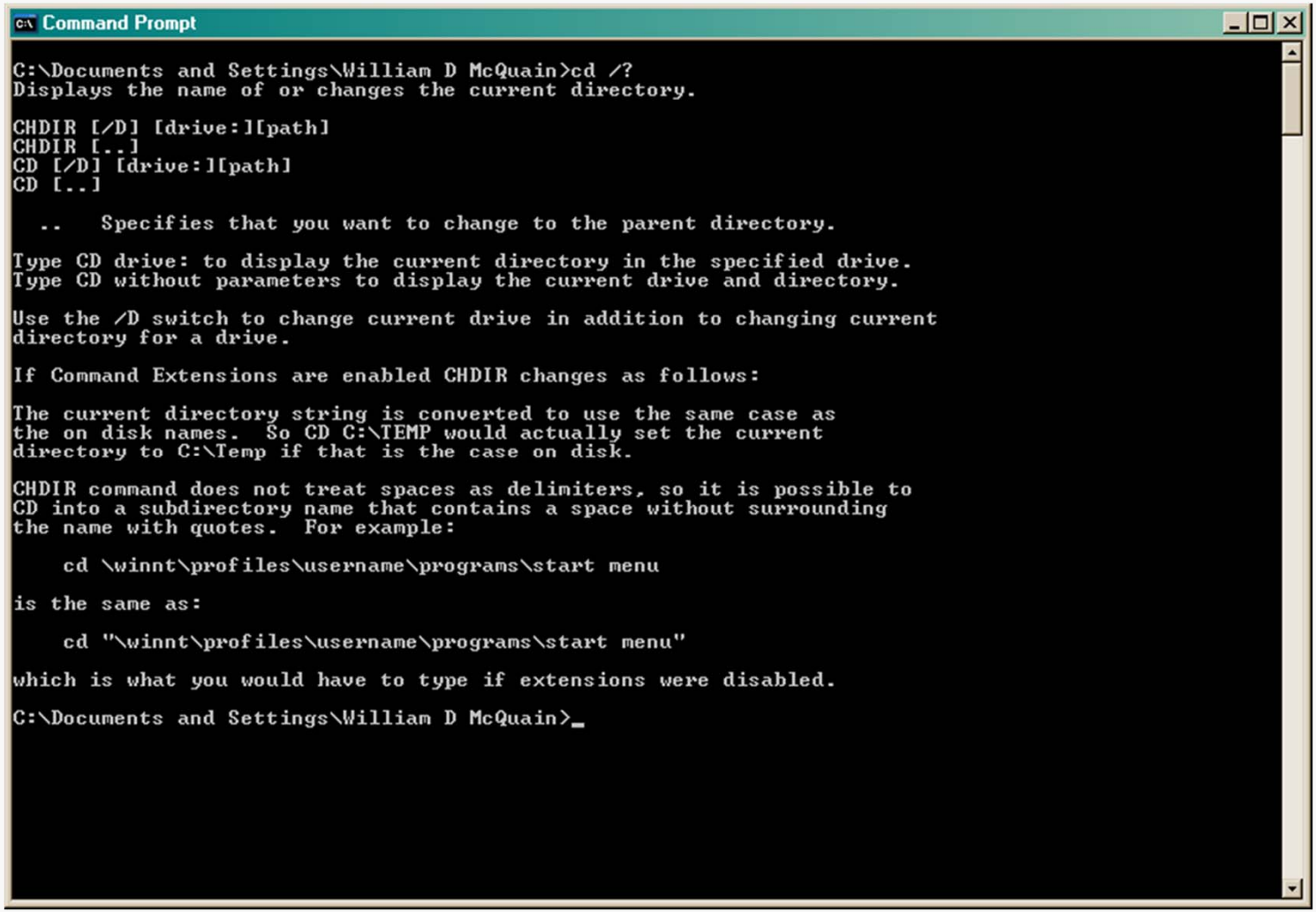
Typing "help" at the command prompt displays a list of shell commands with brief descriptions:



©2010 McQuain

Common Shell Commands

Typing a command followed by "/?" displays a fuller explanation of the command:



©2010 McQuain

Common Shell Commands

The most common shell commands are probably:

`CD`	Displays the name of or changes the current directory.
`CLS`	Clears the screen.
`COPY`	Copies one or more files to another location.
`DEL`	Deletes one or more files.
`DIR`	Displays a list of files and subdirectories in a directory.
`MKDIR`	Creates a directory.
`MOVE`	Moves one or more files from one directory to another directory.
`PATH`	Displays or sets a search path for executable files.
`REN`	Renames a file or files.
`RMDIR`	Removes a directory.

Use the command-line switch `\?` or `help <command>` to get more information about a command.

©2010 McQuain

Command Window Here for Vista/Win7

By default, a command shell opens in your home directory. Navigating from that to another directory is an exercise in forming full and regular paths and using the cd command. It's tedious.

Fortunately, Vista and Win7 make it easy to open a command shell in any directory.

Hold down the Shift key and right-click on the desired folder. The context menu will the include the choice "Open command window here".

©2010 McQuain

Command Window Here for XP

For WinXP:

- Google XP PowerToys
- download and install the Open Command Window Here power toy

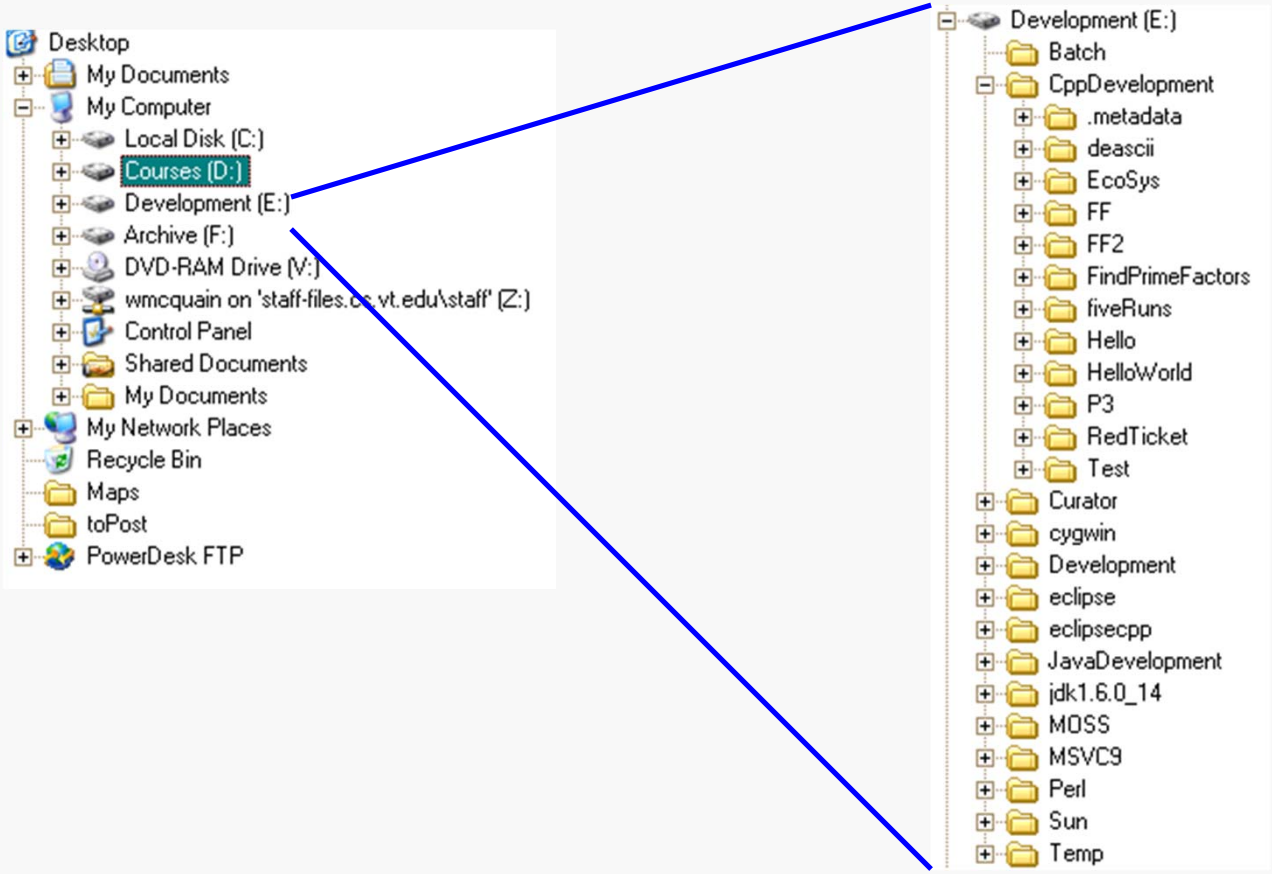
Now, if you right-click a folder in Windows Explorer, you'll have a new option:

©2010 McQuain

Hierarchical File System

You should already be familiar with the basic notion of a hierarchical file system.

We may have a number of logical drives, and separate physical devices:



©2010 McQuain

File System and Paths

Each drive, like E shown on the previous slide is organized logically as a tree of nested directories (or folders).

The top-level directory is called the *root directory* and is denoted by the drive letter, a colon and a back-slash; for example `E:\`

There is a unique *full path* from the root directory to every other directory on the drive.

The full path is formed by concatenating directory names, separated by a back-slash:

```
E:\CppDevelopment\EcoSys\
```

We can also form a *relative path* from one directory to another. The parent of the current directory is denoted by `".."`.

For example:

```
..\Redticket\
```


©2010 McQuain

Command History

The shell stores the most-recently executed commands in a history list.

Use the F7 function key to display a scrollable list of those commands. You can use the up/down arrow keys to navigate the list, and press Enter to repeat the selected command.

You can also use the up/down arrow keys directly at the command prompt to scroll through the previous commands.

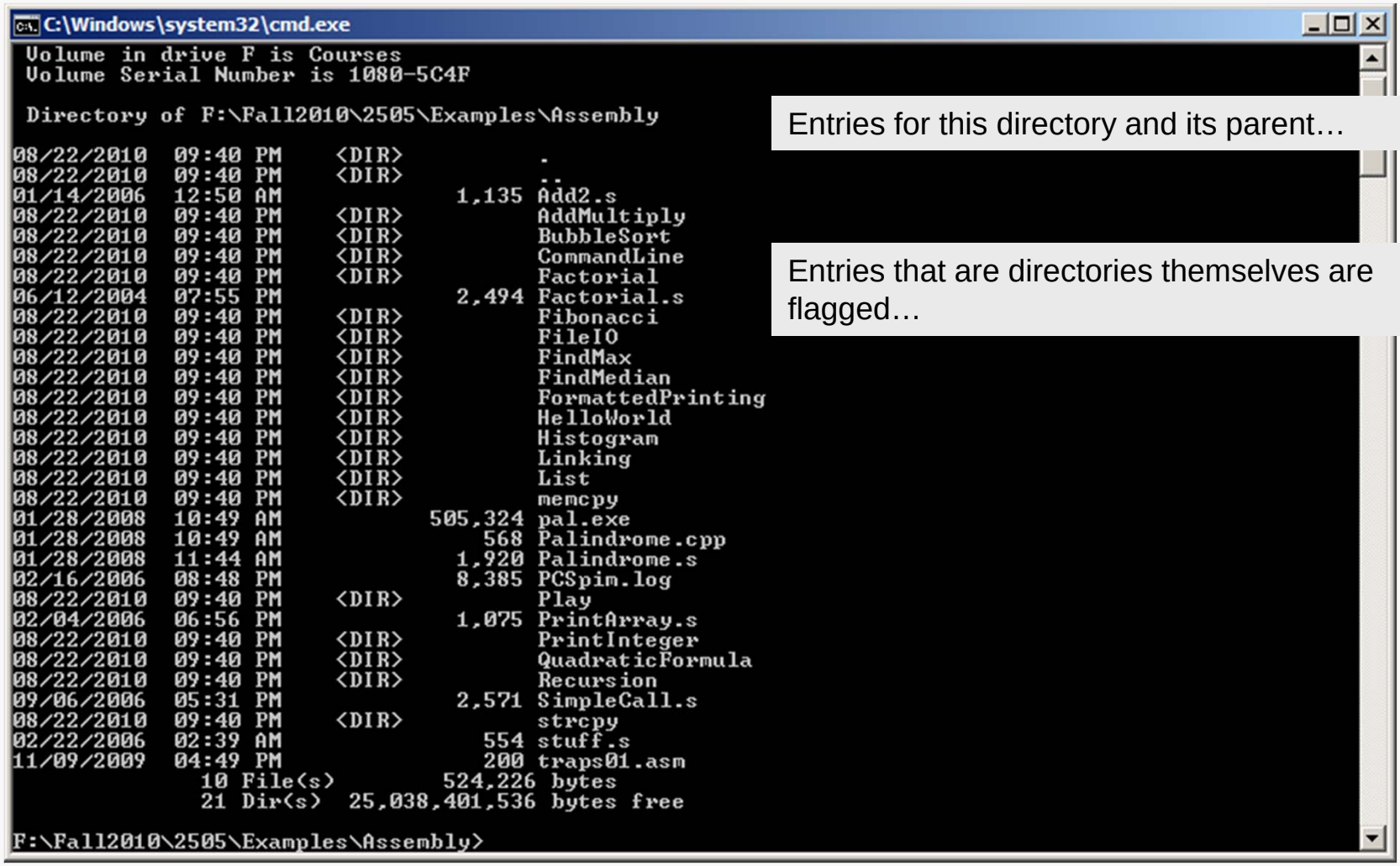
Note that you can use this feature to re-display a previous command and then edit that command to change file names or command-line switches.

The length of the history list can be set through the shell window Properties dialog.

©2010 McQuain

Directory Listing

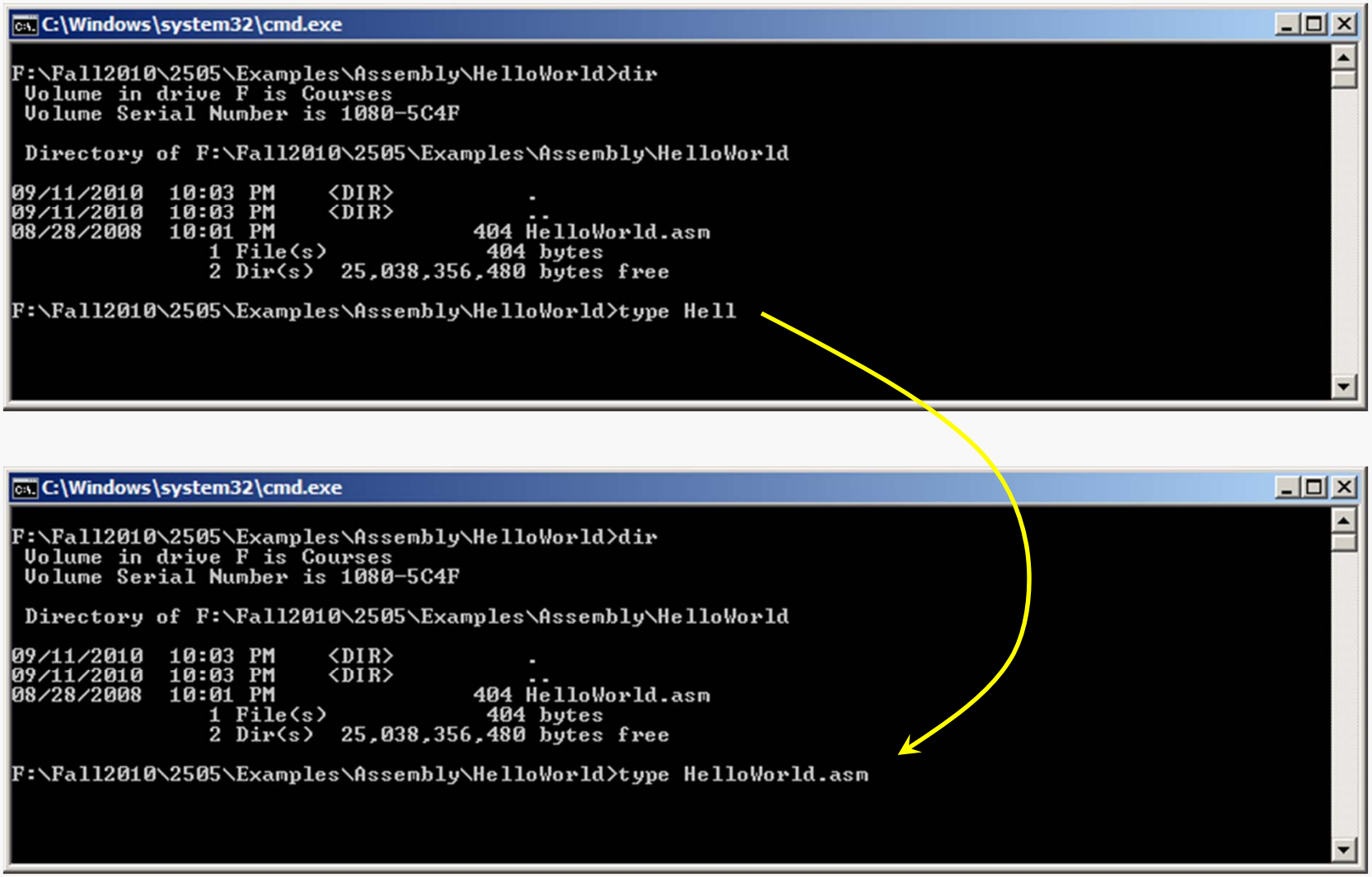
The `DIR` command produces a listing of the files and directories in the current directory:



©2010 McQuain

Tab-completion

If you partially type a file name and then hit the TAB key, the shell (may) complete the name for you:



©2010 McQuain

Running gcc

For the following examples, we assume that your Windows path is set to contain the directory containing the `gcc` executable.

If you are running the Cygwin emulator package, see the following slide for relevant instructions.

As an alternative, you could prefix the invocation of the `gcc` executable with the full path to it; for example:

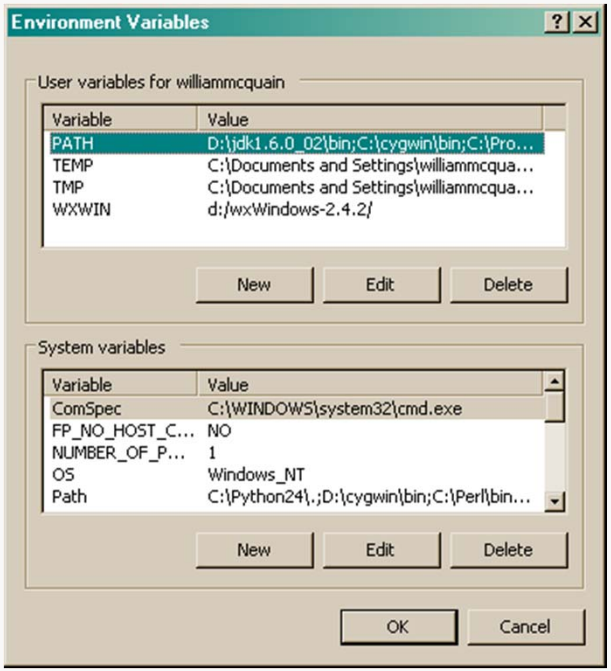
```
C:\Cygwin\bin\gcc-4
```


©2010 McQuain

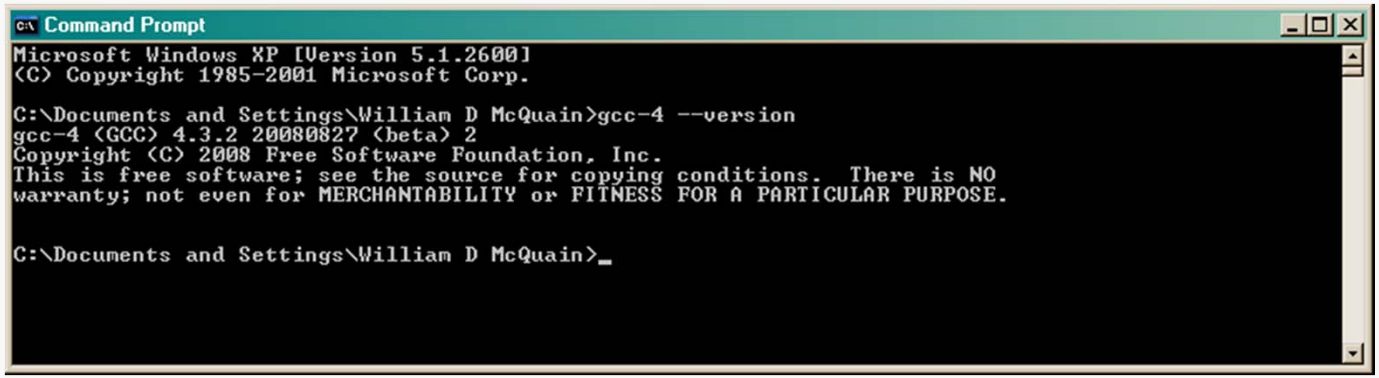
Setting the Windows Path

Cygwin doesn't add itself to the Windows path automatically. Go to Control Panel and run the System applet. Select the Advanced tab, and click on Environment Variables.

Select Path under System variables and add the path to the bin subdirectory of your Cygwin installation. You can see mine listed as the second entry in the Path at right. Path entries are separated by semicolons and are not case-sensitive.



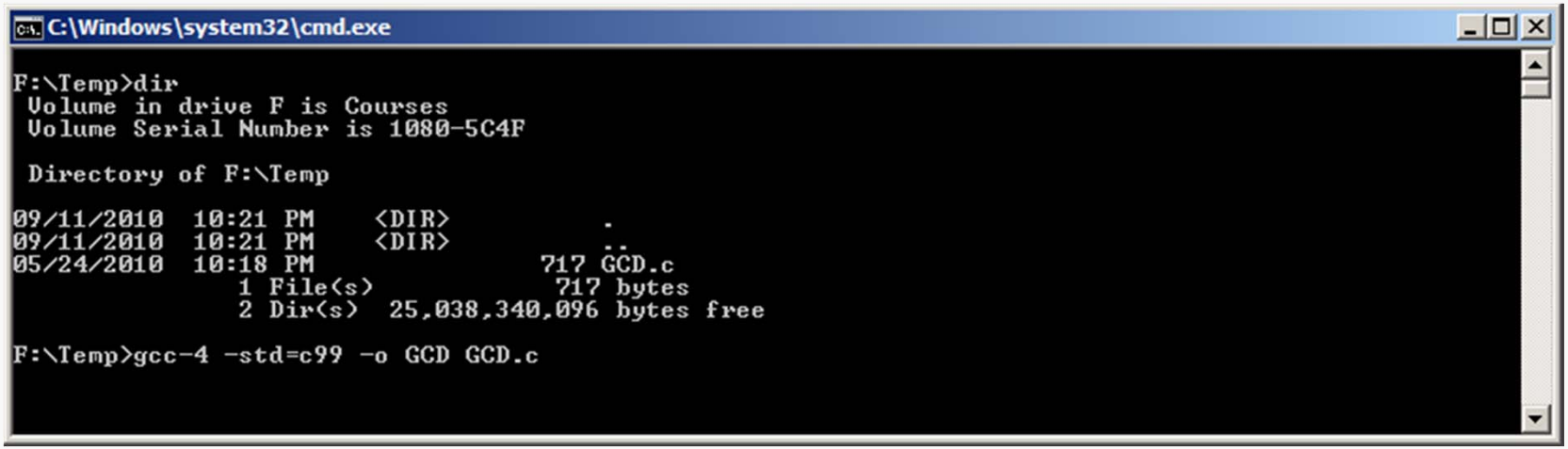
To check your installation (in a very minimal way), open a Windows command shell and see if gcc is recognized:



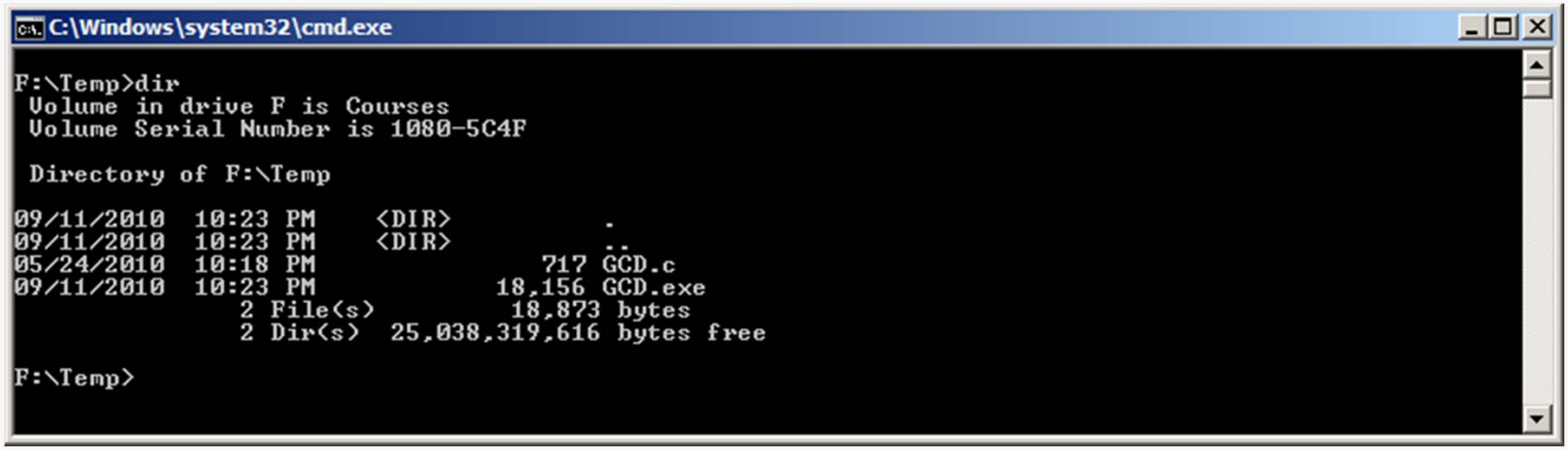
©2010 McQuain

Running gcc

To compile a C program from the directory containing the source file:

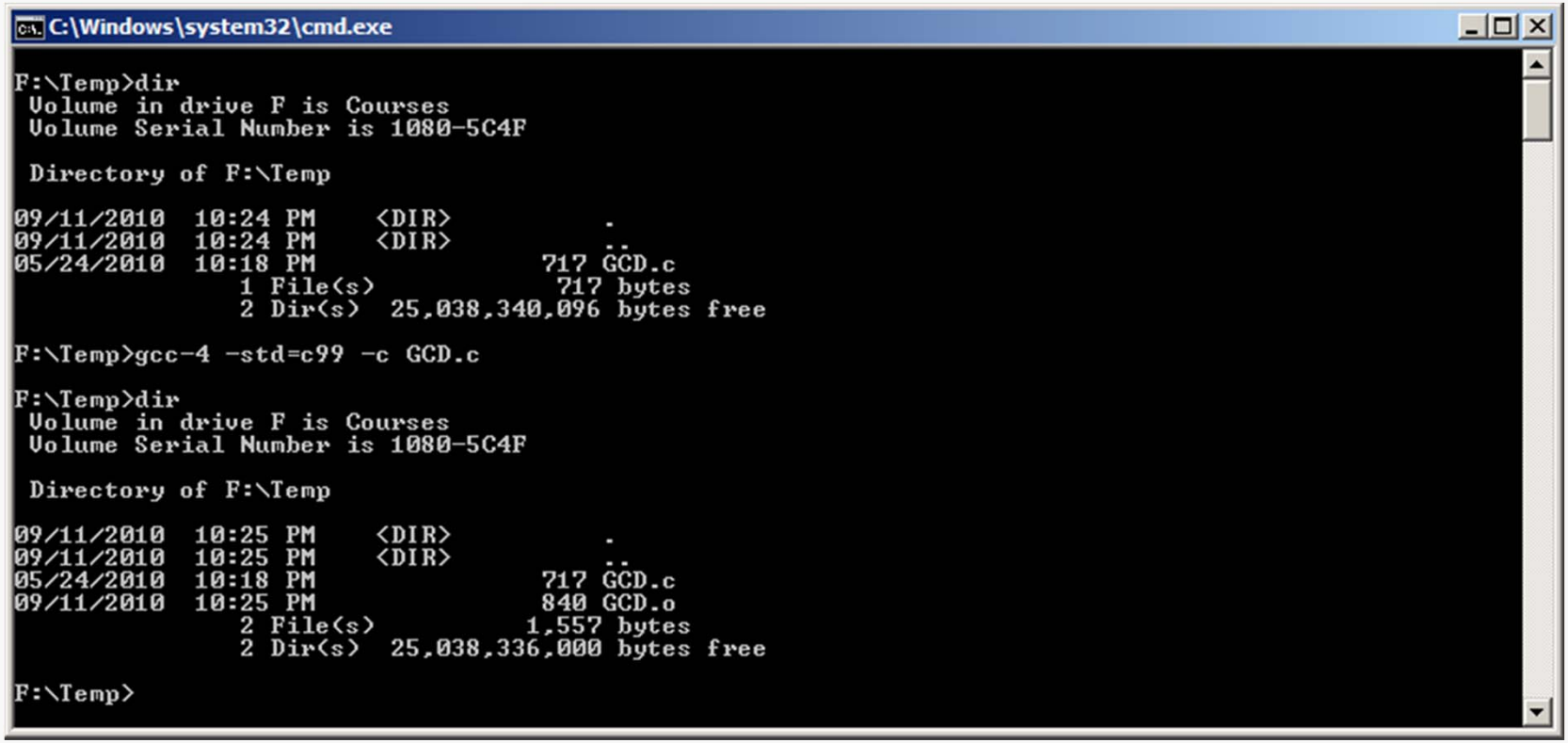


Any error messages will be displayed in the shell window; otherwise an executable appears:



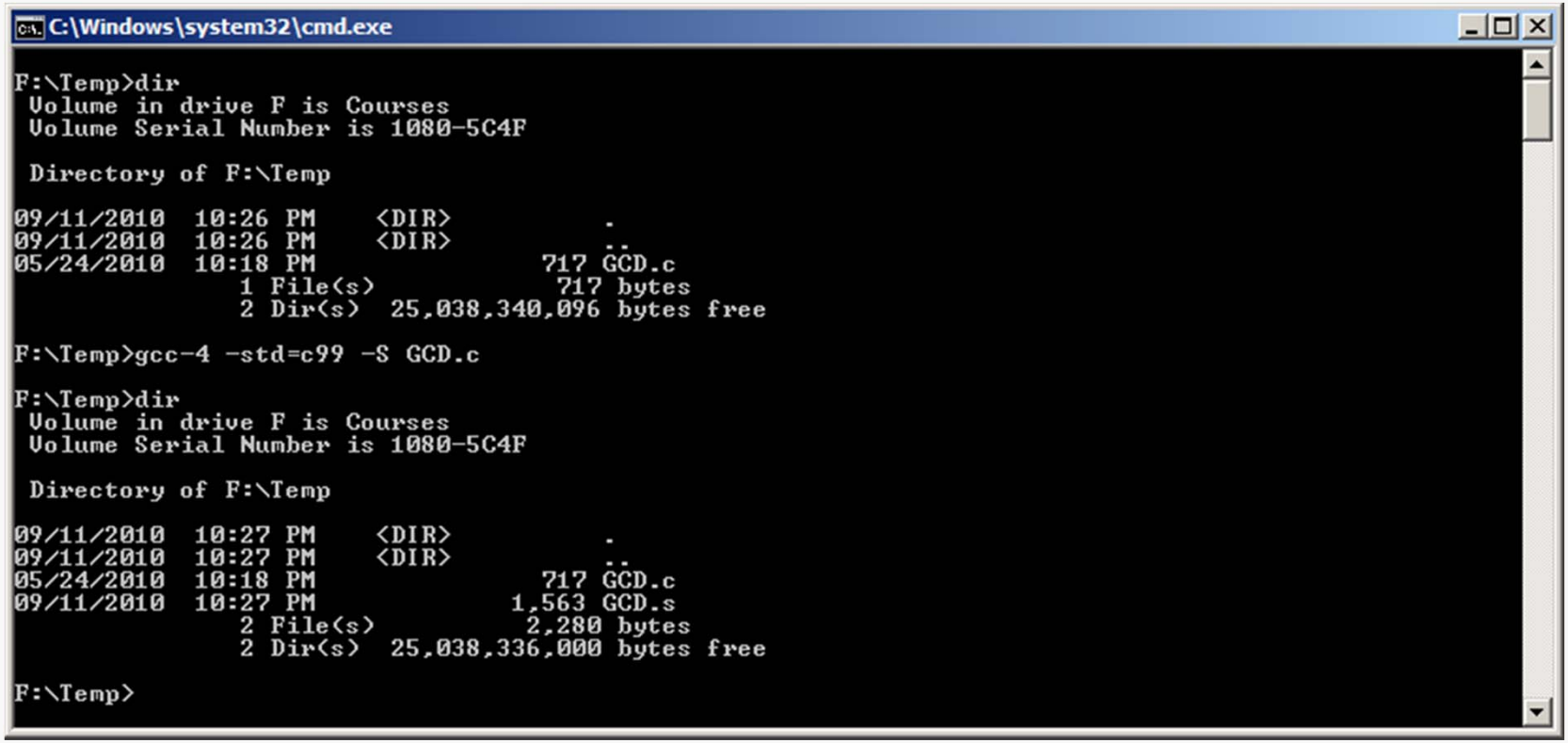
©2010 McQuain

Compile w/o Linking



©2010 McQuain

Compile to (Intel) Assembly Language



©2010 McQuain

Other gcc Options

Some of the most common gcc options:

`-c`
compile and assemble but do not link

`-S`
compile but do not assemble or link; yields an assembly language listing

`-o <name>`
specify name for executable file

`-std=<standard>`
compile to specified standard (`c89`, `c99` most commonly)

`-W`
inhibit all warning messages

`-Werror`
make all warnings be errors

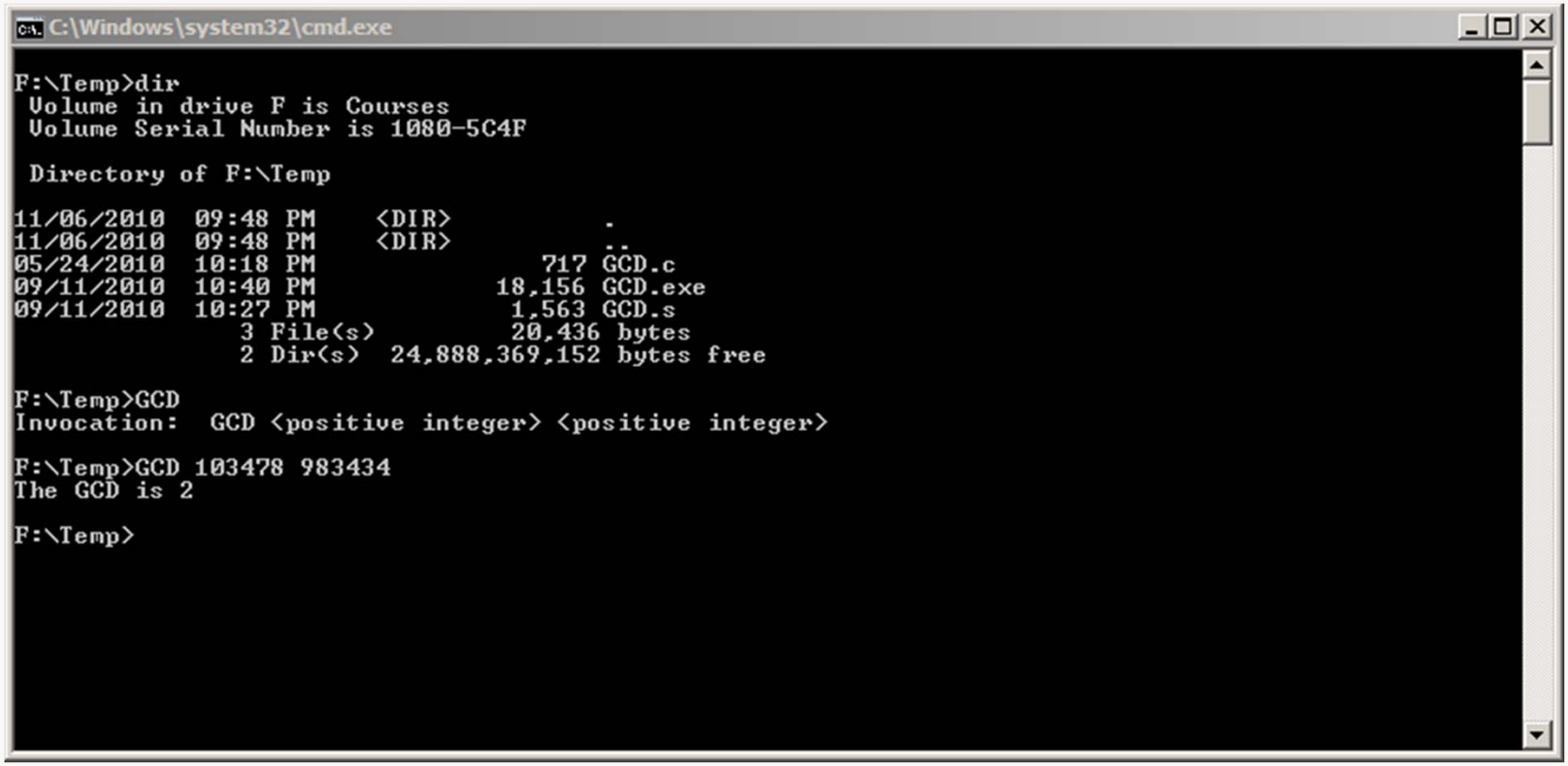
`-Wall`
show more warning messages

`-pedantic`
require strict compliance with the specified standard

©2010 McQuain

Executing a C Program

To execute a program (Windows **exe**cutable file) from the command-line, just type the name of the file:



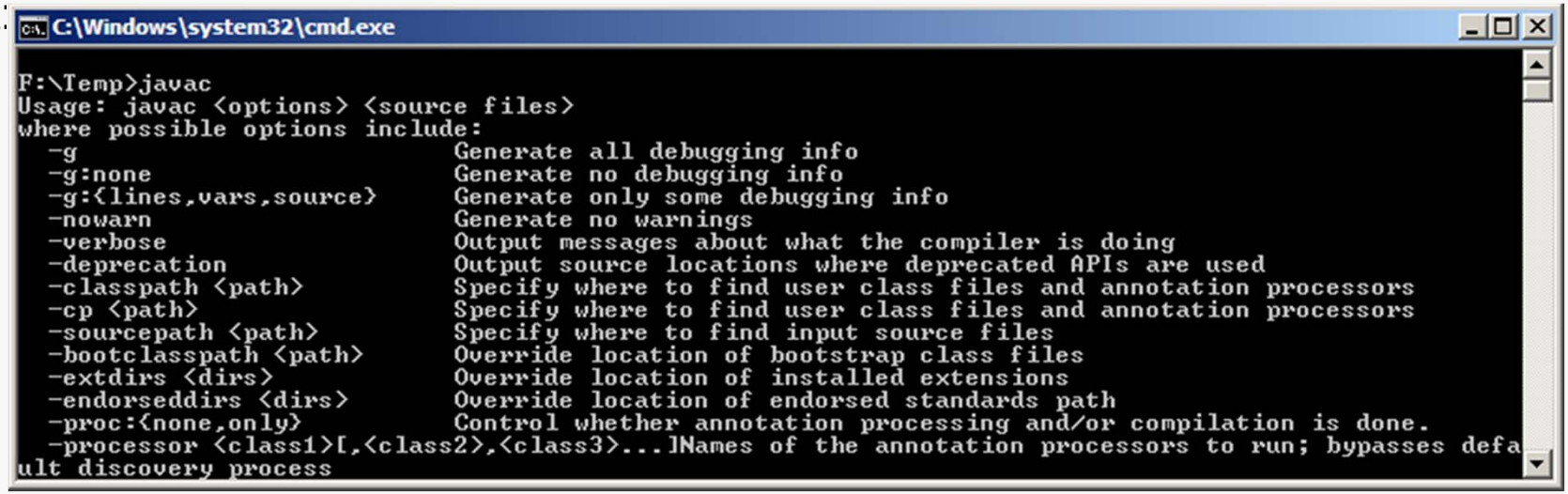
Of course, in some cases the program may require you also supply parameters from the command-line, as shown above…

©2010 McQuain

Compiling Java Programs

To compile a Java program, you must use a Java compiler; Sun provides one as part of the Java Development Kit (JDK), named `javac`.

You can test whether the compiler is in the Windows path by attempting to execute it:



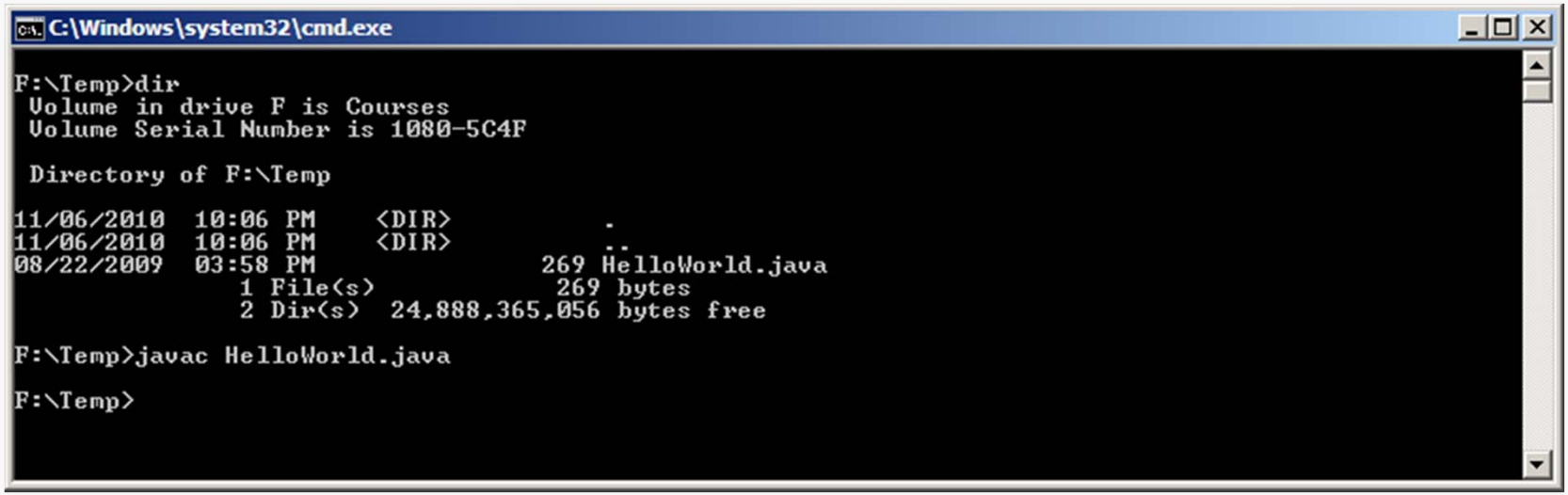
If everything is set up correctly, you should see a display similar to the one shown above.

If not, you must add the appropriate directory to your Windows path.

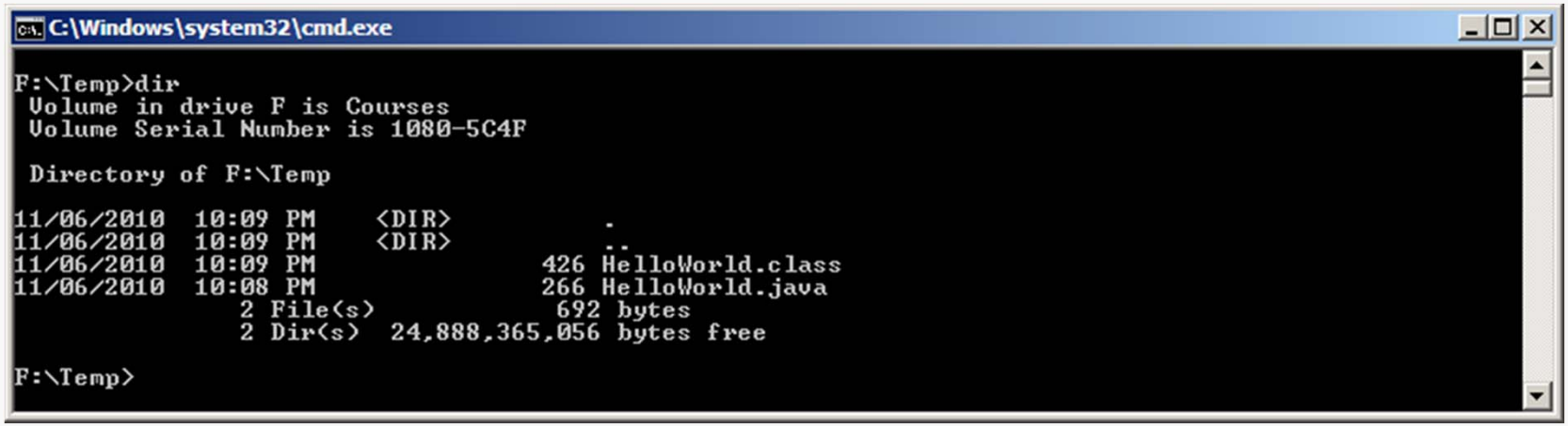
©2010 McQuain

Compiling Java

To compile a single-file Java program from the directory containing the source file:



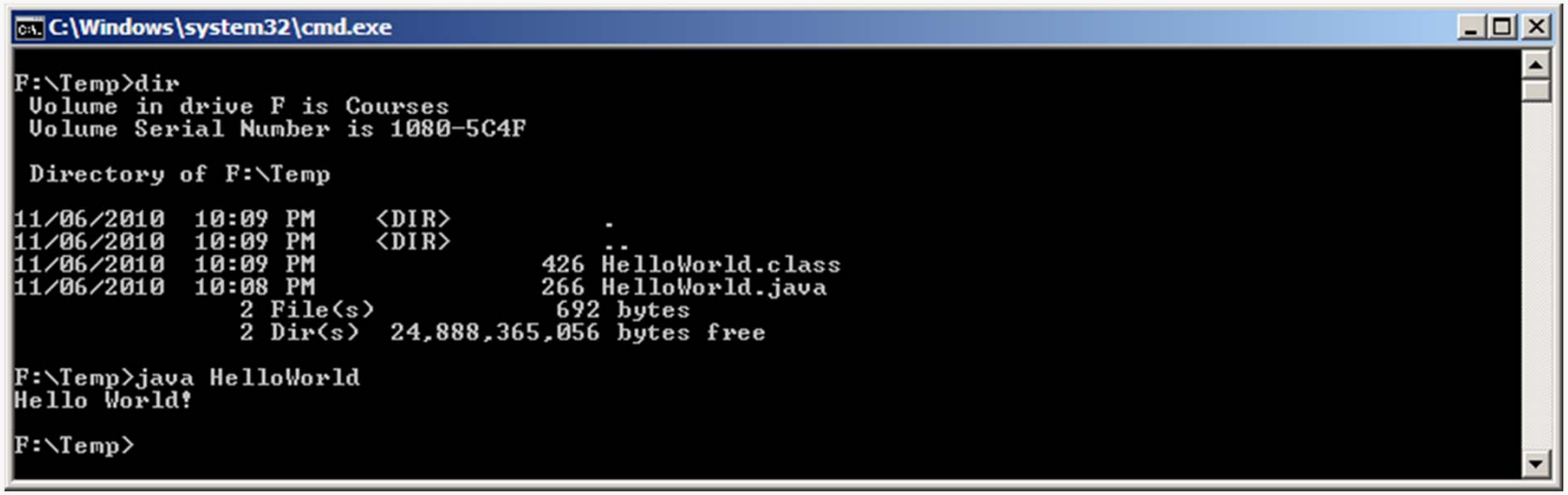
Any error messages will be displayed in the shell window; otherwise a Java class file appears:



©2010 McQuain

Executing a Java Program

To execute a Java program from the command-line:



Note that the Java interpreter is named `java`.

Any error messages or output will appear in the command shell window.

The general rule is that you invoke the Java interpreter (`java`) on the class file that contains the implementation of `static void main()`.

©2010 McQuain

A Java Program with Packages

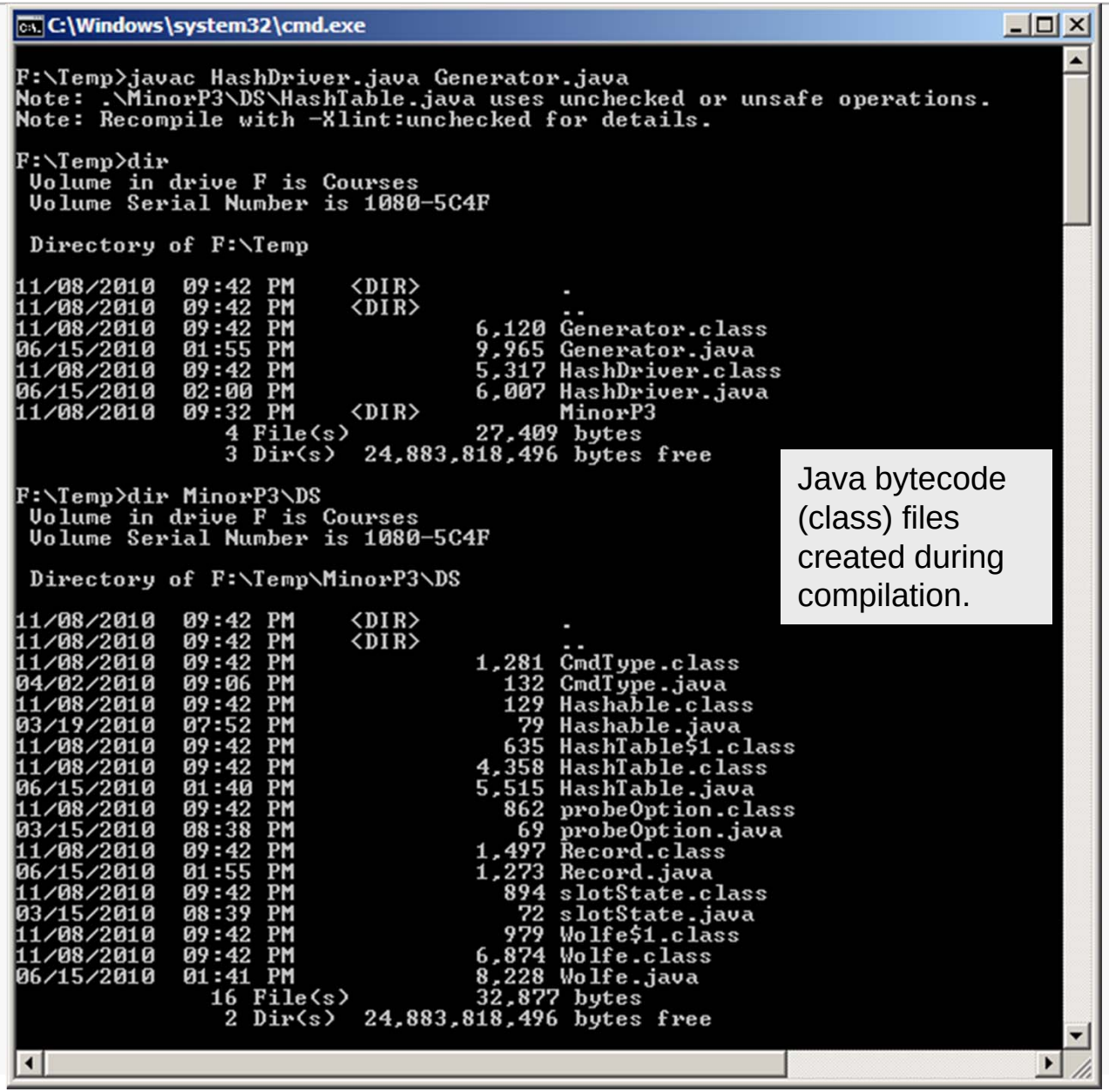
Suppose we have a multi-file Java program:

```
.(base directory)
  |-- HashDriver.java
  |-- Generator.java
  |-- MinorP3/
        |-- DS\
              |-- CmdType.java
              |-- Hashable.java
              |-- HashTable.java
              |-- probeOption.java
              |-- Record.java
              |-- slotState.java
              |-- Wolfe.java
```


©2010 McQuain

Compiling a Java Program with Packages

To compile, it is sufficient to specify the top-level files, since they, directly or indirectly, reference all the package files:

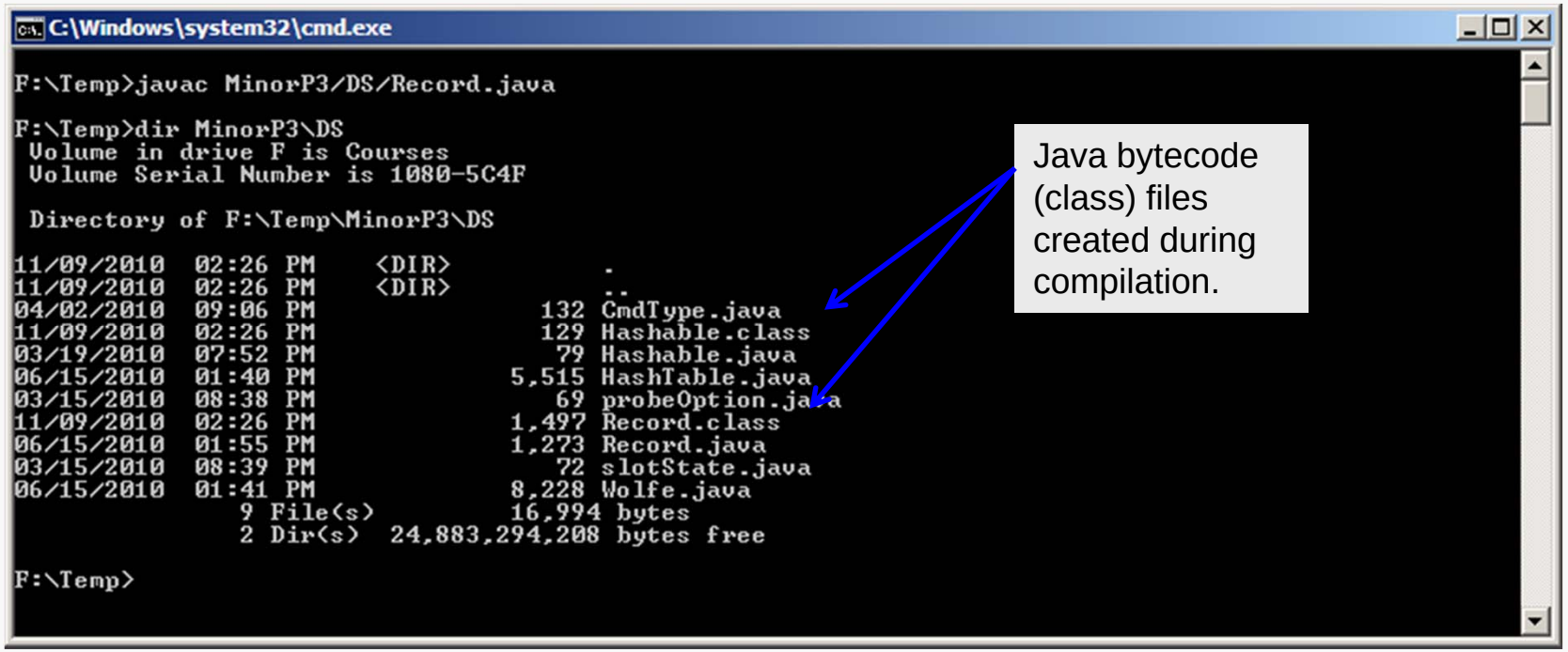


Java bytecode (class) files created during compilation.

©2010 McQuain

Compiling a Java Program with Packages

To compile a single class in the package:

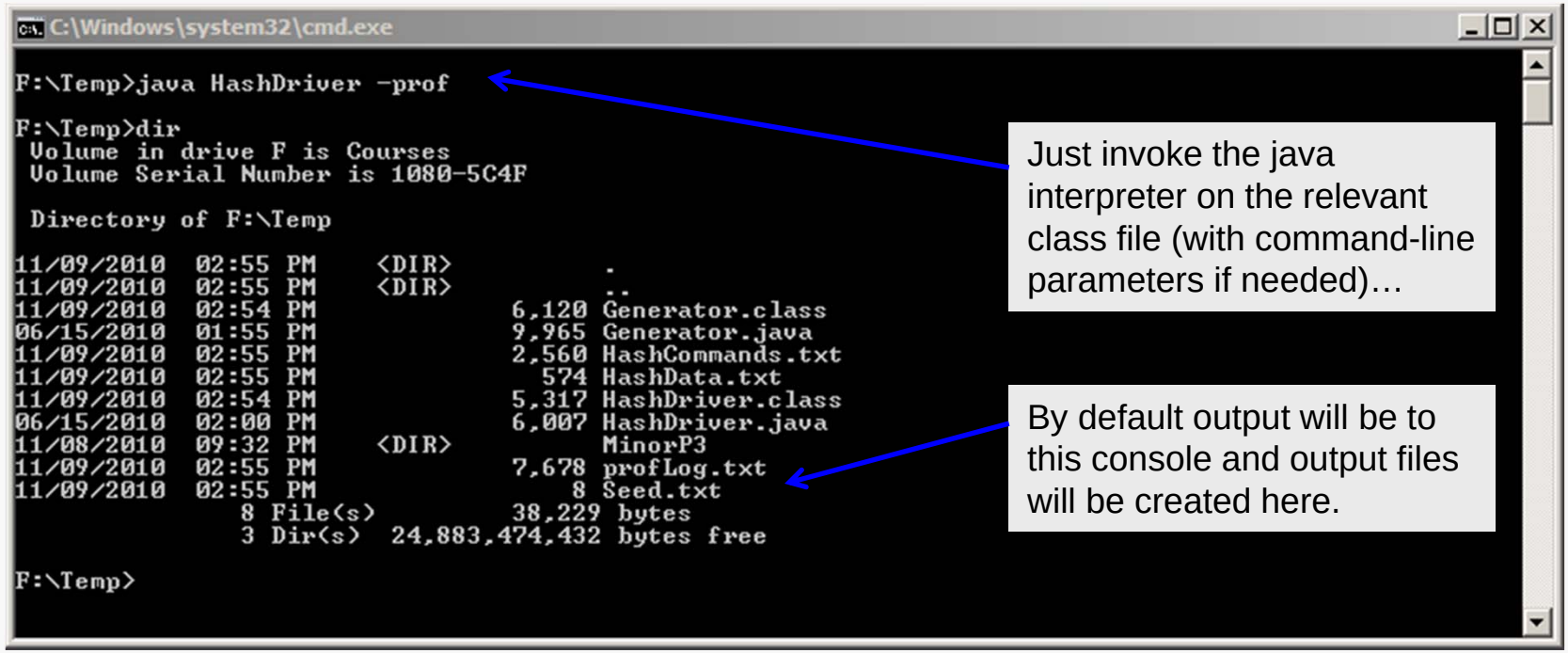


`Hashable.java` was also compiled because it is referenced in `Record.java`.

©2010 McQuain

Executing a Java Program with Packages

If the "main" class is in the top-level directory (not in the package):



If the "main" class is in the package, invoke the interpreter from the top-level directory but specify the "path" to the main class file:

```
java MinorP3/DS/MyDriver
```


©2010 McQuain