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.
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.
Typing "help" at the command prompt displays a list of shell commands with brief descriptions:
Typing a command followed by "/?" displays a fuller explanation of the command:
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.
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 include the choice "Open command window here".
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:
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:
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\n```

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

For example:

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

Entries for this directory and its parent...

Entries that are directories themselves are flagged...
Tab-completion

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

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

```
F:\Temp>dir
Volume in drive F is Courses
Volume Serial Number is 1080-5C4F
Directory of F:\Temp
09/11/2010 10:21 PM   <DIR>          ..
05/24/2010 10:18 PM   717  GCD.c
1 File(s)           717 bytes
2 Dir(s)            25,038,340,096 bytes free
F:\Temp>gcc -std=c99 -o GCD GCD.c
```

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

```
F:\Temp>dir
Volume in drive F is Courses
Volume Serial Number is 1080-5C4F
Directory of F:\Temp
09/11/2010 10:23 PM   <DIR>         ..
05/24/2010 10:18 PM   717  GCD.c
18,156  GCD.exe
2 File(s)           18,873 bytes
2 Dir(s)            25,038,319,616 bytes free
F:\Temp>
```
Compile to (Intel) Assembly Language
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
Executing a C Program

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

```
F:\Temp>dir
Volume in drive F is Courses
Volume Serial Number is 1080-5C4F
Directory of F:\Temp
11/06/2010  09:48 PM  <DIR>  .
11/06/2010  09:48 PM  <DIR>  ...
05/24/2010  10:18 PM  717 GCD.c
09/11/2010  10:27 PM  18,156 GCD.exe
09/11/2010  10:27 PM  1,563 GCD.s
3 File(s)  20,436 bytes
2 Dir(s)  24,888,369,152 bytes free

F:\Temp>GCD
Invocation:  GCD <positive integer> <positive integer>
F:\Temp>GCD 103478 983434
The GCD is 2
F:\Temp>
```

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

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

```
F:\Temp>dir
Volume in drive F is Courses
Volume Serial Number is 1080-5C4F
Directory of F:\Temp
11/06/2010 10:06 PM <DIR>         .
11/06/2010 10:06 PM <DIR>         ..
08/22/2009 03:58 PM <DIR>         HelloWorld.java
 1 File(s)          269 bytes
 2 Dir(s)           269 bytes free

F:\Temp>javac HelloWorld.java
F:\Temp>
```

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

```
F:\Temp>dir
Volume in drive F is Courses
Volume Serial Number is 1080-5C4F
Directory of F:\Temp
11/06/2010 10:09 PM <DIR>         .
11/06/2010 10:09 PM <DIR>         ..
11/06/2010 10:09 PM <DIR>         HelloWorld.class
11/06/2010 10:08 PM <DIR>         HelloWorld.java
 2 File(s)           426 bytes
 2 Dir(s)           692 bytes free

F:\Temp>
```
Executing a Java Program

To execute a Java program from the command-line:

![Command prompt window with Java program execution]

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()`.
Compiling a Multi-file Java Program

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:
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: