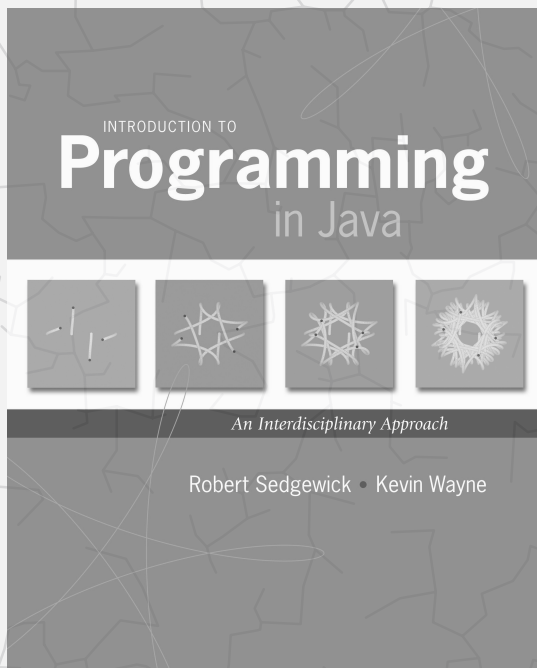


<http://introc.cs.princeton.edu>

COMMAND-LINE INTERFACE

- ▶ *brief overview*
- ▶ *live demo*



<http://introc.cs.princeton.edu>

COMMAND-LINE INTERFACE

- ▶ *brief overview*
- ▶ *live demo*

Today's plan: command-line interface

Brief overview.

- IDE vs. command line.
- Files and file systems.

Live demo.

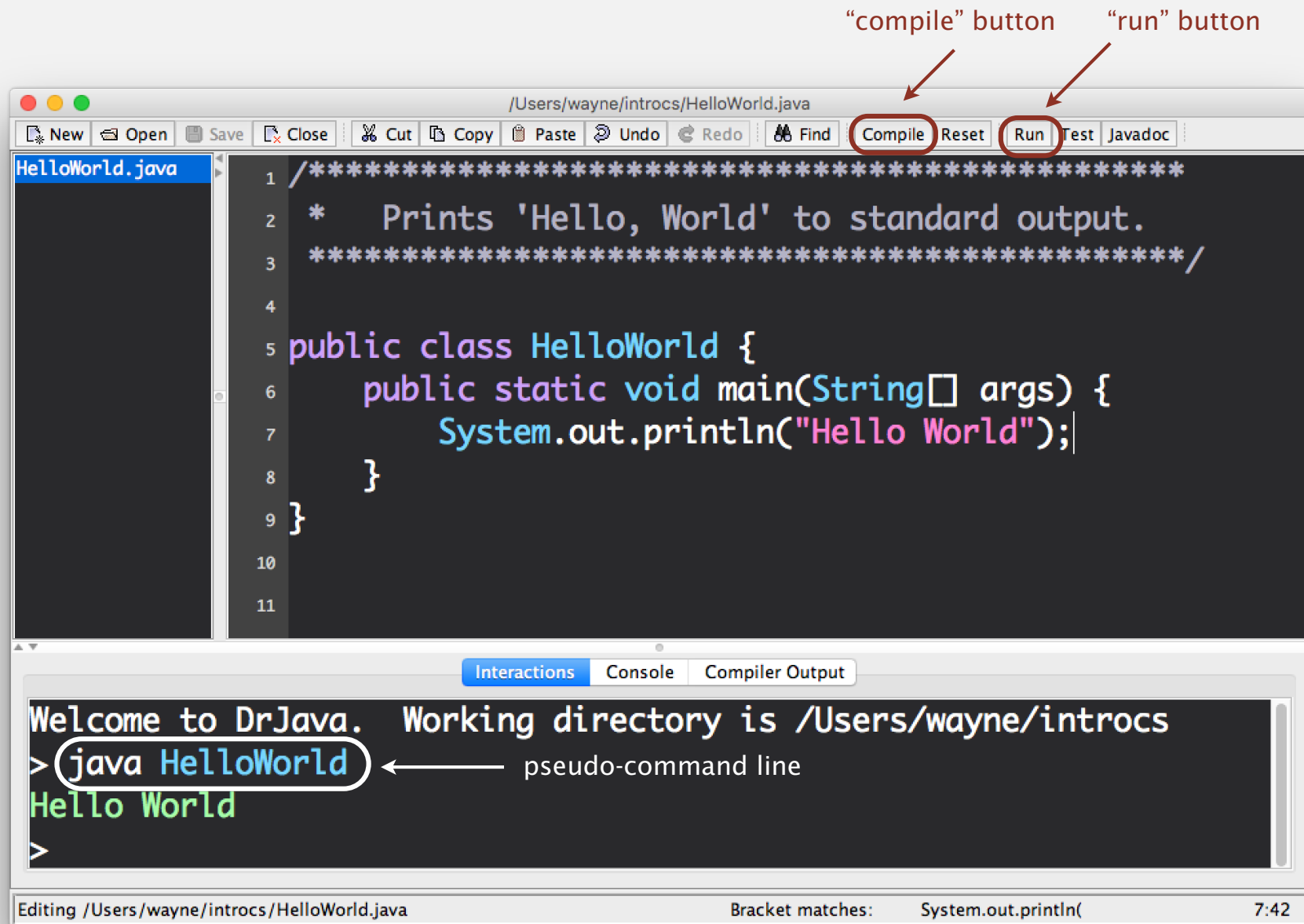
- Navigating the file system.
- Java commands (including accessing `stdlib.jar`).
- Redirection and piping of standard input and output.

Note: you will need CLI for the Assignment 2.

Integrated development environment (IDE)

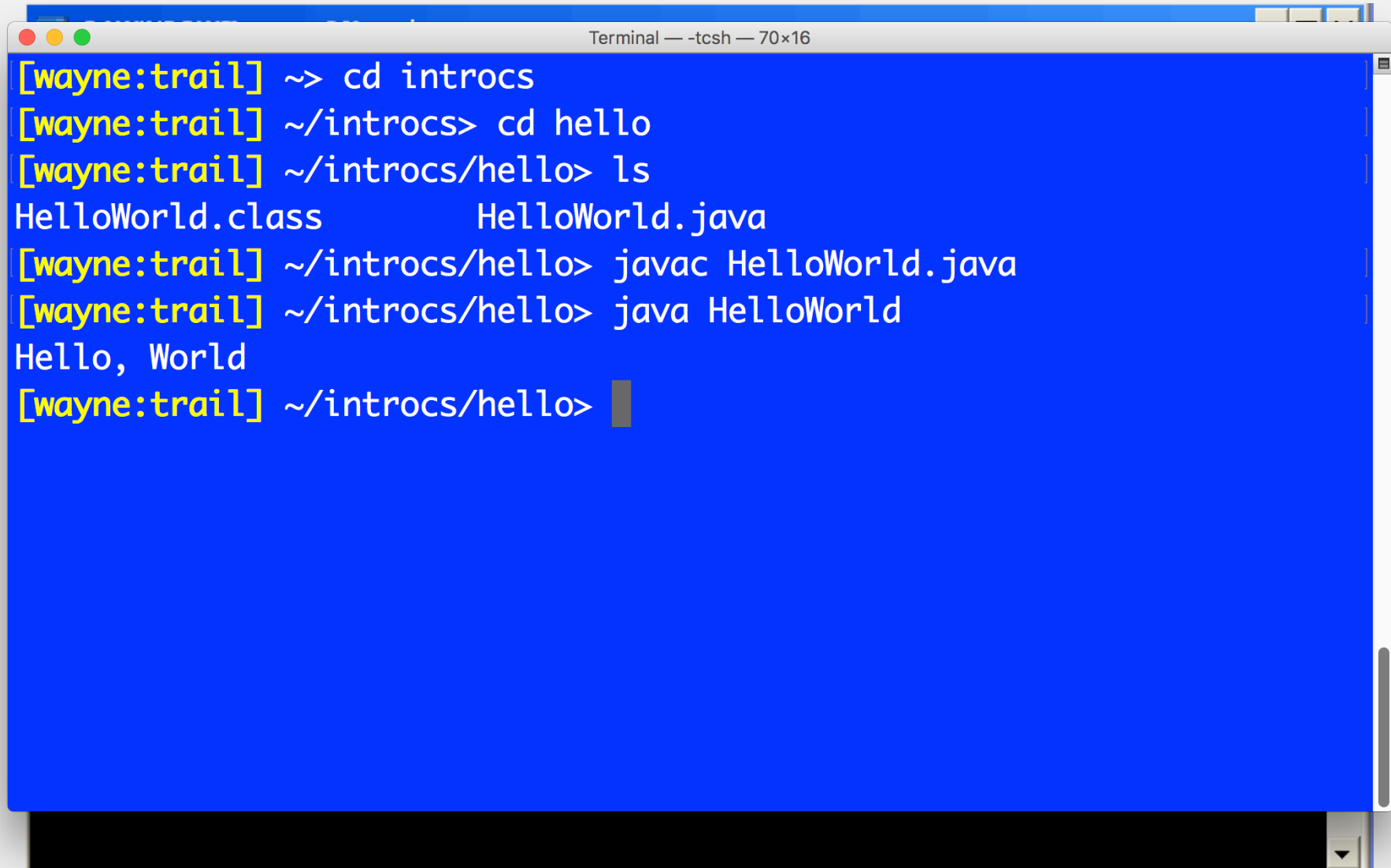


IDE. App designed for developing software.



Command-line interface

Command line. App where you type commands.

A terminal window with a blue background and a title bar that reads "Terminal — -tcsh — 70x16". The terminal shows a series of commands and their outputs. The prompt is always "[wayne:trail]". The commands and outputs are: "cd introcs", "cd hello", "ls" (output: "HelloWorld.class", "HelloWorld.java"), "javac HelloWorld.java", "java HelloWorld" (output: "Hello, World"), and a final prompt with a cursor.

```
[wayne:trail] ~> cd introcs
[wayne:trail] ~/introcs> cd hello
[wayne:trail] ~/introcs/hello> ls
HelloWorld.class      HelloWorld.java
[wayne:trail] ~/introcs/hello> javac HelloWorld.java
[wayne:trail] ~/introcs/hello> java HelloWorld
Hello, World
[wayne:trail] ~/introcs/hello> █
```

Convergence: Linux, OS X, and Windows



Linux

(September 1991)

```
chealer@vinci:/usr/share/doc/bash$ export LC_ALL=C
chealer@vinci:/usr/share/doc/bash$ cd ~chealer/
chealer@vinci:~$ ls
Cloutier Ido      Musique logs      skolo sources
Desktop Mes images boston ncix.png smb4k vieux
chealer@vinci:~$ #why is there color when calling ls without arguments?
chealer@vinci:~$ which ls
/bin/ls
chealer@vinci:~$ ${!!}
$(which ls)
Cloutier Ido      Musique logs      skolo sources
Desktop Mes images boston ncix.png smb4k vieux
chealer@vinci:~$ type ls #`ls` doesn't just run /bin/ls
ls is aliased to `ls --color=auto'
chealer@vinci:~$ echo $PS1
${debian_chroot:+(${debian_chroot})}\u@h:\w$
chealer@vinci:~$ sh
sh-3.1$ echo $PS1
\s-\w$
sh-3.1$ echo $BASH_VERSION
3.1.17(1)-release
sh-3.1$ ls
Cloutier Ido      Musique logs      skolo sources
Desktop Mes images boston ncix.png smb4k vieux
sh-3.1$ echo $SHELLOPTS # ls isn't an alias in POSIX mode
braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor:posix
sh-3.1$ kill
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill
-l [sigspec]
sh-3.1$ /bin/kill 6> killerror # collect stdout and stderr of $ /bin/kill: in ki
llerror
sh-3.1$ wc -l !$
wc -l killerror
7 killerror
sh-3.1$ type kill # kill doesn't just run /bin/kill, even in POSIX mode.
kill is a shell builtin
sh-3.1$ !$ -n 9 $$ # OK, kill self
kill -n 9 $$ # OK, kill self
Killed
chealer@vinci:~$
```

Convergence: Linux, OS X, and Windows



(March 2001)



Terminal

Command-line access to UNIX.

OS X is built on an industry-standard UNIX foundation. The Terminal application allows you to access the complete UNIX environment using standard commands, tools, and scripting languages. Terminal includes numerous shortcuts that will save you time. For example, you can copy and paste text commands, and if you drag files or folders into Terminal, the correct file path appears on the command line. You can customize Terminal with your favorite fonts and background colors, including transparent, as well as manage multiple sessions from a single window using tabs.

A screenshot of a Terminal window titled 'Terminal -- top -- 90x40'. The window shows the output of the 'top' command, which displays system statistics and a list of running processes. The system statistics at the top include: Processes: 76 total, 2 running, 1 stuck, 73 sleeping... 290 threads; Load Avg: 0.06, 0.25, 0.19; CPU usage: 1.09% user, 4.72% sys, 93.40% idle; SharedLibs: mux = 0, resident = 45M code, 708K data, 3840K linked; MemRegions: mux = 9539, resident = 270M + 10M private, 12M shared; PhysMem: 395M wired, 352M active, 62M inactive, 755M used, 269M free; VM: 5591M + 377M 409205(10) pageins, 144510(0) pageouts. The process list below shows columns for PID, COMMAND, %CPU, TIME, #TH, #PRTS, #PMEM, #PRVT, #SHRD, #SIZE, and #VSIZE. The top process is 'screencapt' with PID 5516, using 0.0% CPU and 0.00.02 seconds of time.

Convergence: Linux, OS X, and Windows



Here's how Windows 10's Ubuntu-based Bash shell will actually work

Aimed at developers, Bash brings a complete Linux command line to Windows, but only the command line.

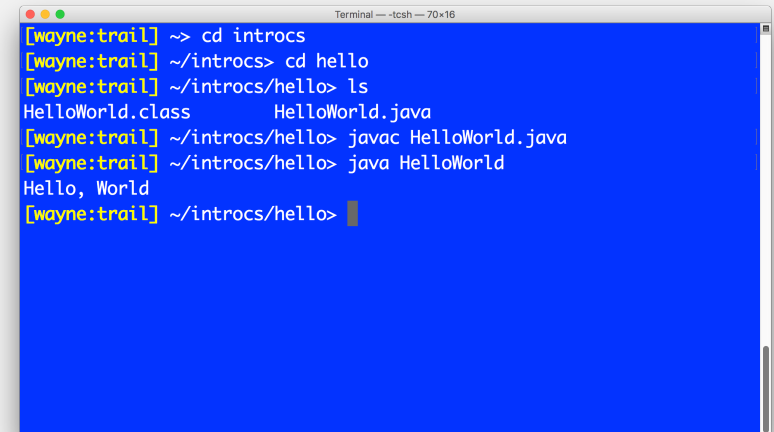


Bash coming to Windows

Software for program development: tradeoffs

Command line advantages.

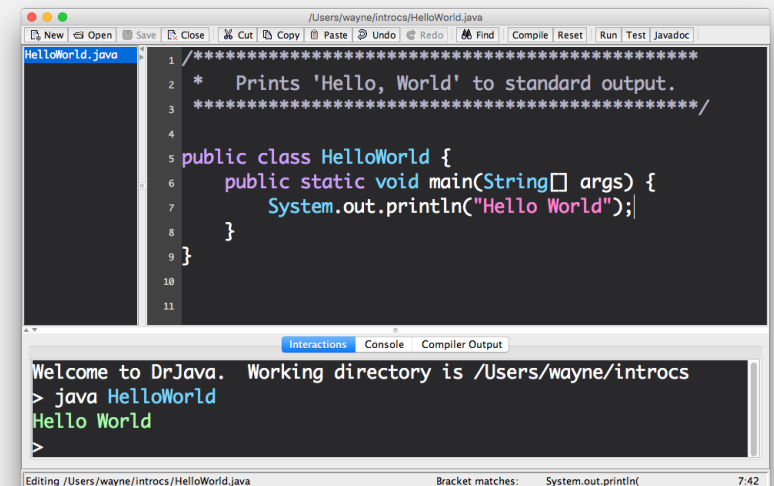
- More control over system.
- Approach works with any language.
- Easy to automate tasks via scripting.



```
Terminal -- tcsh -- 70x16
[wayne:trail] ~> cd introcs
[wayne:trail] ~/introcs> cd hello
[wayne:trail] ~/introcs/hello> ls
HelloWorld.class      HelloWorld.java
[wayne:trail] ~/introcs/hello> javac HelloWorld.java
[wayne:trail] ~/introcs/hello> java HelloWorld
Hello, World
[wayne:trail] ~/introcs/hello>
```

IDE advantages.

- More intuitive for novices.
- Language-specific features.



```

1  /*****
2  *   Prints 'Hello, World' to standard output.
3  *****/
4
5  public class HelloWorld {
6      public static void main(String[] args) {
7          System.out.println("Hello World");
8      }
9  }
10
11

Welcome to DrJava. Working directory is /Users/wayne/introcs
> java HelloWorld
Hello World
>
```

This course. Use IDE to edit and compile; use command line to execute.

Beyond. Many other platforms embrace command line.



Files

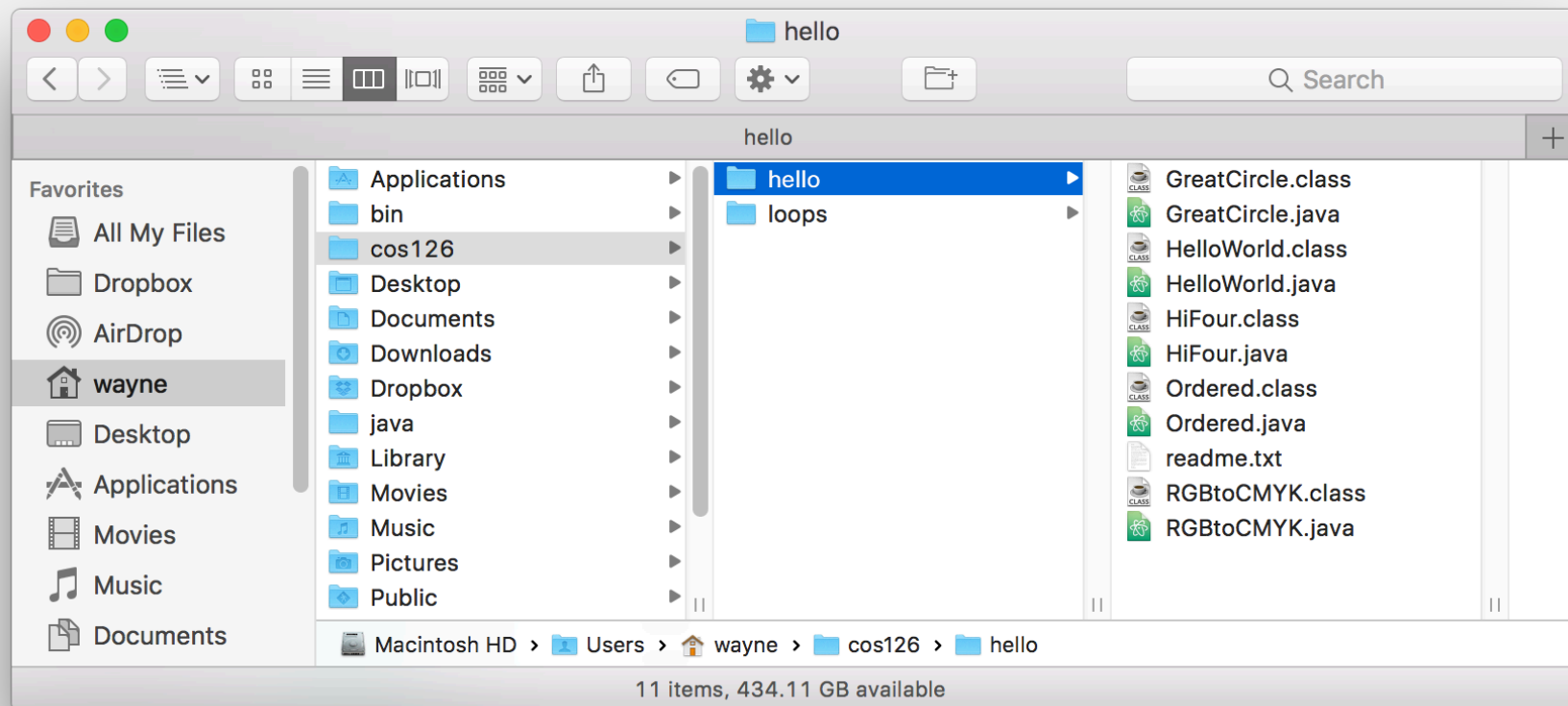
File. Sequence of bits stored on a computer.

Filename. String that uniquely identifies a file.

Mac OS X. `/Users/wayne/cos126/he11o/He11oWorld.java`

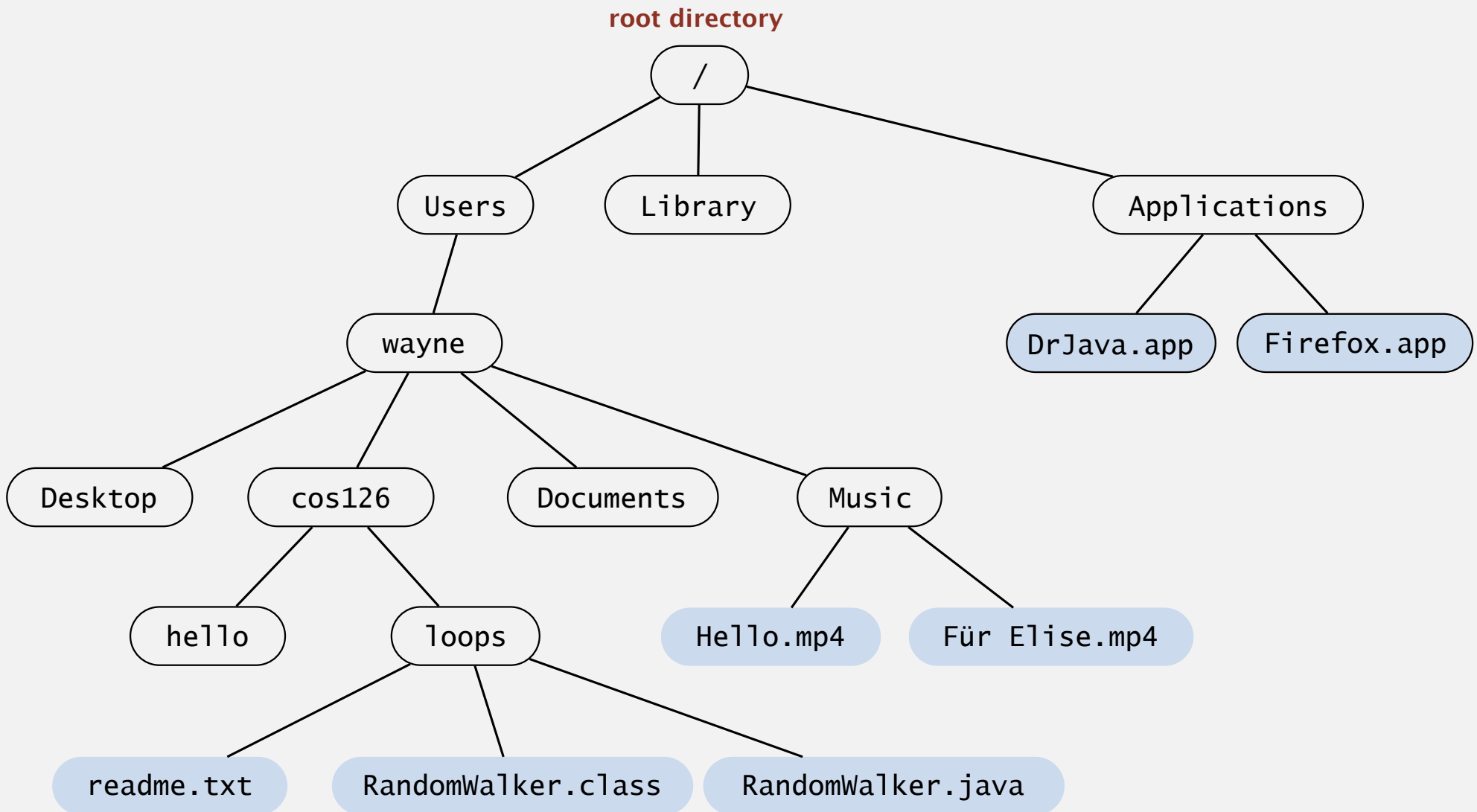
Windows. `C:\Users\wayne\cos126\he11o\He11oWorld.java`

← note: forward vs.
← backward slash

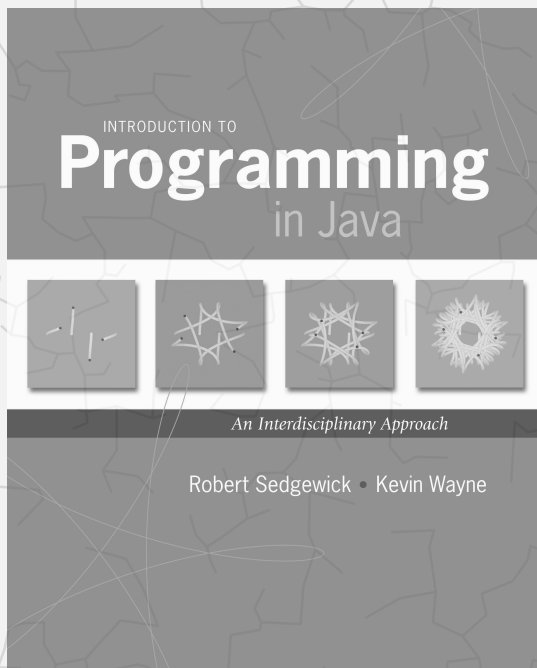


File system hierarchy

Directory (folder). Collection of files and other folders.



`/Users/wayne/cos126/loops/RandomWalker.java`



<http://introc.cs.princeton.edu>

COMMAND-LINE INTERFACE

- ▶ *brief overview*
- ▶ *live demo*

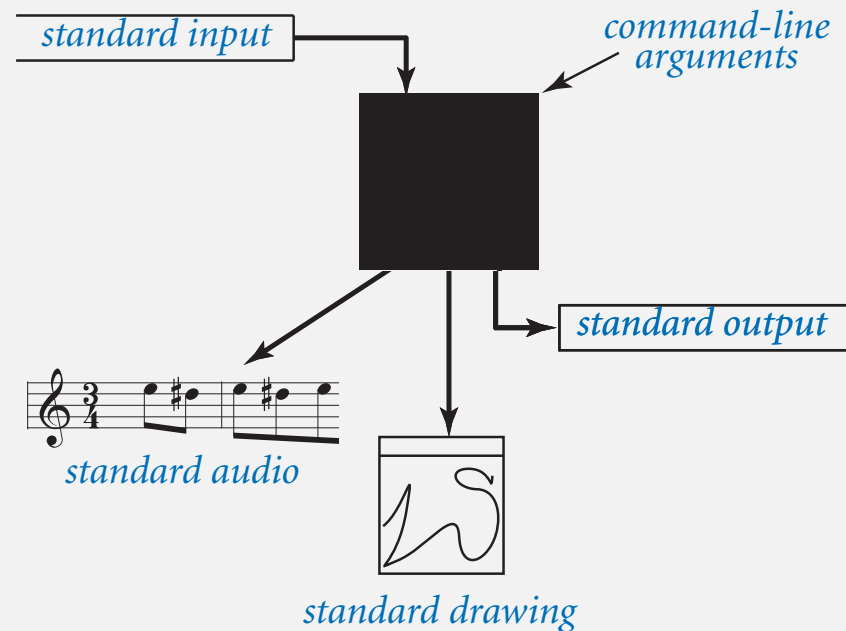
Live demo of file system commands

Standard input and output abstractions

Command-line arguments. Provide arguments (strings) to a program.

Standard output stream. Abstraction for writing output (default = screen).

Standard input stream. Abstraction for reading input (default = keyboard).



Live demo of standard input and output commands

Navigating the file system

Action	Mac OS X Terminal	Windows Command Prompt
<i>print current directory</i>	<code>pwd</code>	<code>cd</code>
<i>list contents of current directory</i>	<code>ls</code>	<code>dir</code>
<i>move down one subdirectory</i>	<code>cd introcs</code>	
<i>move up one subdirectory</i>	<code>cd ..</code>	
<i>move down multiple subdirectories</i>	<code>cd introcs/hello</code>	<code>cd introcs\hello</code>
<i>move to specified directory</i>	<code>cd /Users/wayne</code>	<code>cd \Users\wayne</code>
<i>copy, move, remove, make directory, remove directory</i>	<code>cp mv rm</code> <code>mkdir rmdir</code>	<code>copy move del</code> <code>mkdir rmdir</code>
<i>cycle through command history</i>	<code><Up Arrow> <Down Arrow></code>	
<i>autocomplete</i>	<code>javac Hello<Tab></code>	

Java commands

Action	Mac OS X Terminal	Windows Command Prompt
<i>compile Java program</i>	<code>javac HelloWorld.java</code>	
<i>execute Java program</i>	<code>java HelloWorld</code>	
<i>compile Java program (with our standard libraries)</i>	<code>javac-introcs HelloWorld.java</code>	
<i>execute Java program (with our standard libraries)</i>	<code>java-introcs HelloWorld</code>	
<i>file globbing</i>	<code>javac *.java</code>	
<i>Java compiler version</i>	<code>javac -version</code>	
<i>Java runtime version</i>	<code>java -version</code>	

Command-line arguments, redirection, and piping

Action	Mac OS X Terminal	Windows Command Prompt
<i>command-line arguments</i>	java Program arg1 arg2	
<i>redirect standard input</i>	java Program < input.txt	
<i>redirect standard input</i>	java Program > output.txt	
<i>piping</i>	java Program1 java Program2	
<i>view contents, one screenful at a time</i>	more < input.txt java Program more	
<i>combinations</i>	java Program1 < input.txt more java Program < input.txt > output.txt java Program1 arg1 java Program2 arg2	
<i>terminate program</i>	<Ctrl-C>	
<i>signal end-of-file (when typing standard input)</i>	<Ctrl-D>	<Ctrl-Z>