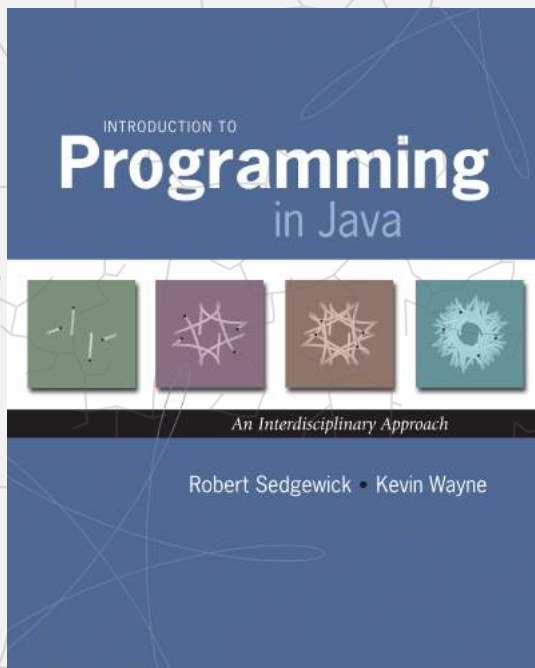


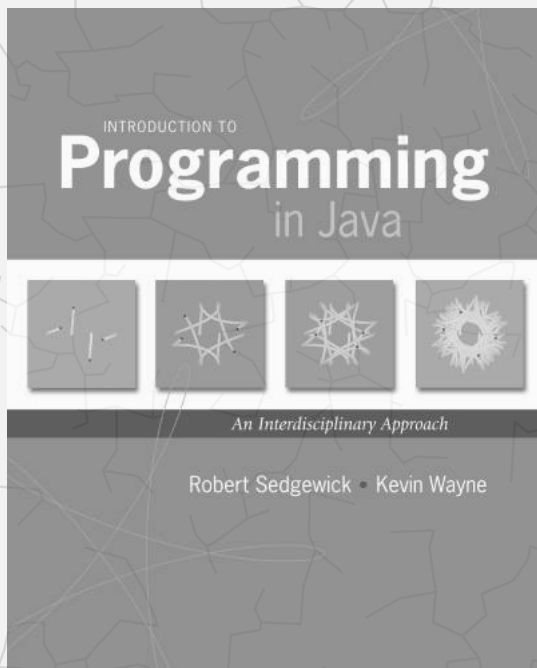
# COMMAND-LINE INTERFACE

---

- ▶ *brief overview*
- ▶ *live demo*



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



<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.

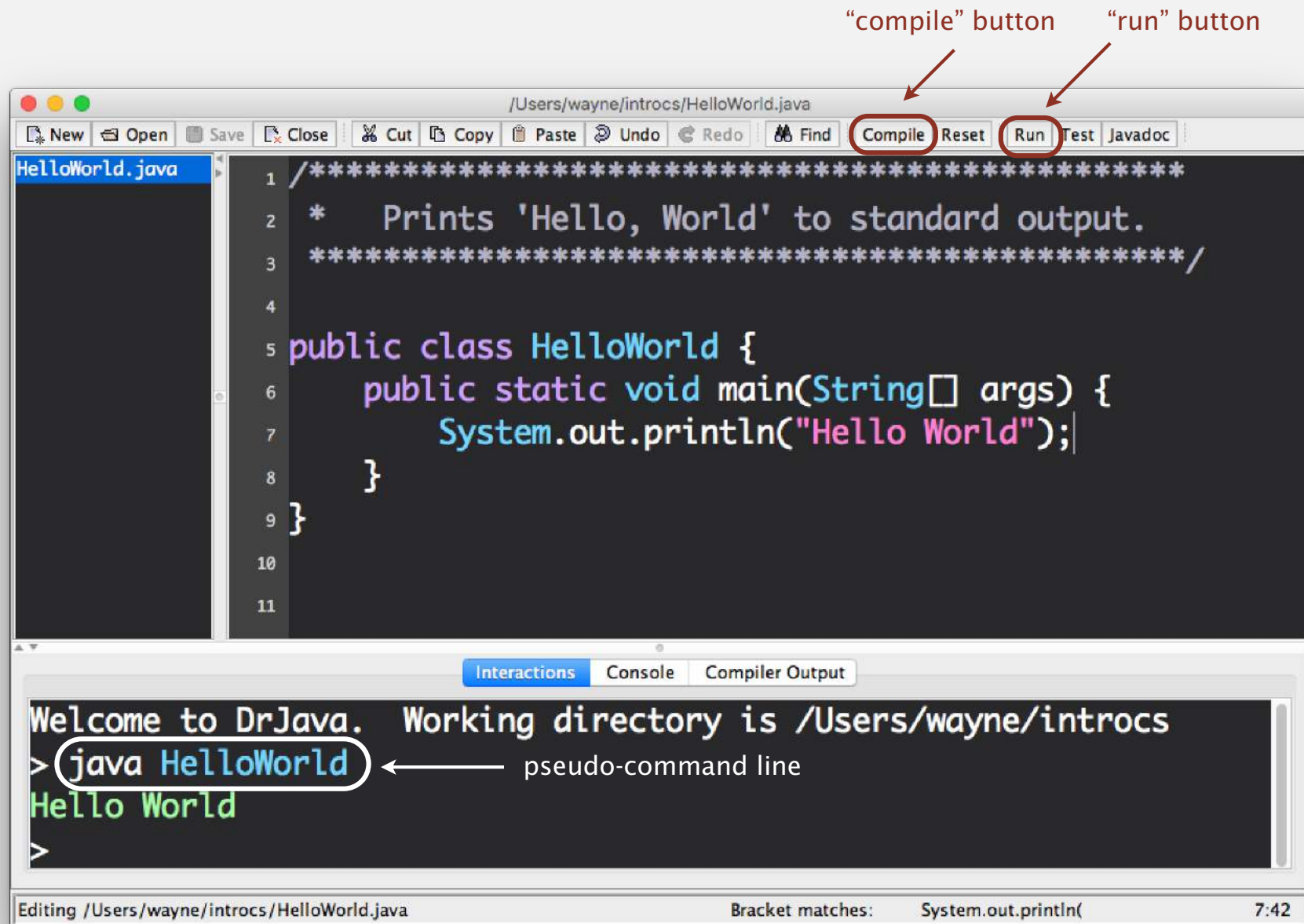
- Navigating the file system.
- Java commands (including accessing `stdlib.jar`).
- Redirection and piping of standard input and output.

**Note:** you will need CLI for the Assignment 2.

# Integrated development environment (IDE)



IDE. App designed for developing software.



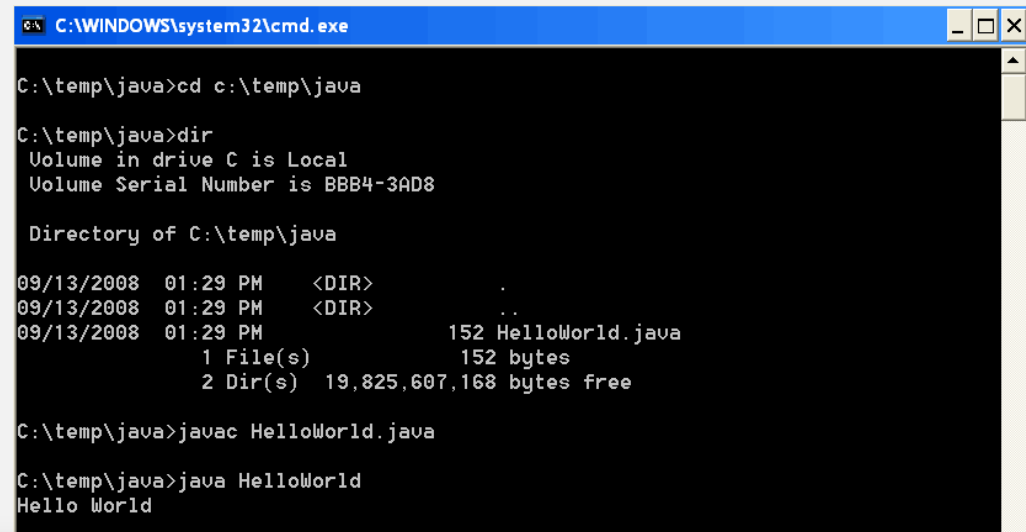
# Command-line interface

---

**Virtual Terminal.** App where you type text commands.

**Shell.** A program that obeys text commands in a terminal.

**Command Line.** Place where you enter shell commands.



```
C:\WINDOWS\system32\cmd.exe

C:\temp\java>cd c:\temp\java

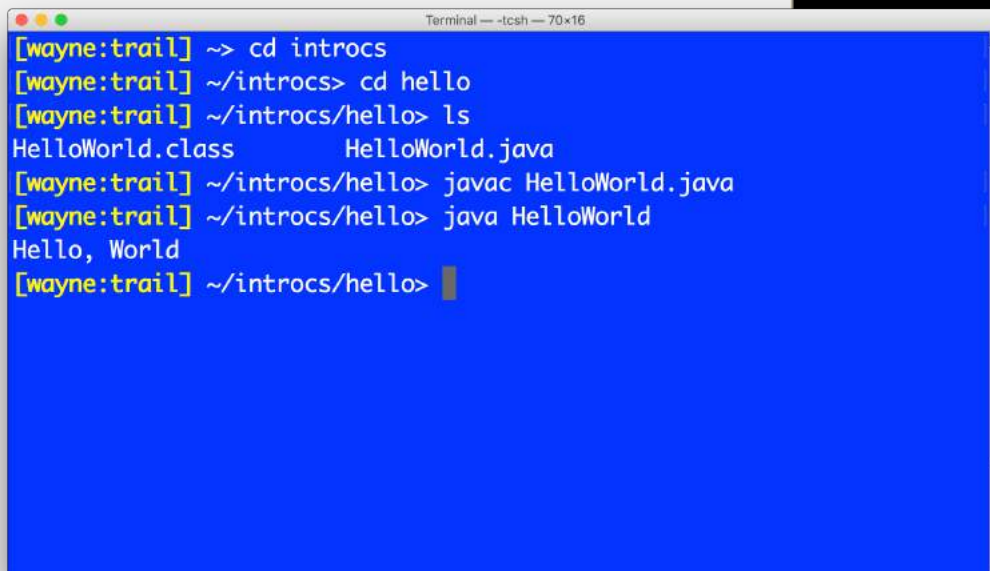
C:\temp\java>dir
Volume in drive C is Local
Volume Serial Number is BBB4-3AD8

Directory of C:\temp\java

09/13/2008  01:29 PM    <DIR>          .
09/13/2008  01:29 PM    <DIR>          ..
09/13/2008  01:29 PM                152 HelloWorld.java
               1 File(s)                152 bytes
               2 Dir(s)  19,825,607,168 bytes free

C:\temp\java>javac HelloWorld.java

C:\temp\java>java HelloWorld
Hello World
```



```
Terminal — tcsh — 70x16

[wayne:trail] ~> cd intros
[wayne:trail] ~/intros> cd hello
[wayne:trail] ~/intros/hello> ls
HelloWorld.class      HelloWorld.java
[wayne:trail] ~/intros/hello> javac HelloWorld.java
[wayne:trail] ~/intros/hello> java HelloWorld
Hello, World
[wayne:trail] ~/intros/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 $SHELLDPTS # ls isn't an alias in POSIX mode
braceexpand:emacs:hashall:histexpand:history:interactive:comments:monitor:posix
sh-3.1$ 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 running the 'top' command. The window title is 'Terminal -- top -- 90x40'. The output shows system statistics and a list of processes. The first few lines of the process list are:

```
PID COMMAND %CPU TIME #TH #PRTS #MEMG1 #PRVT #RMBD #SIZE #VSIZE
5116 screencast 0.0% 0:00.02 1 30* 100 612K* 3900K* 2412K* 171M*
5589 top 5.4% 0:00.00 1 19 29 900K 100K 1572K 10M
5499 bash 0.0% 0:00.00 1 14 19 256K 704K 900K 10M
5490 login 0.0% 0:00.01 1 17 55 344K 260K 1116K 10M
5480 bash 0.0% 0:00.00 1 14 19 240K 704K 892K 10M
5407 login 0.0% 0:00.01 1 17 55 344K 260K 1116K 10M
5454 Terminal 0.0% 0:04.04 5 180 221 4200K 10M 15M 242M
5440 edcorbar 0.0% 0:00.25 3 67 72 1250K 4950K 3020K 31M
5440 edcorbar 0.0% 0:00.22 3 51 30 675K 4932K 2320K 31M
5420 sshd 0.0% 0:00.00 2 21 26 204K 104K 696K 10M
5426 iTunesMtp 0.0% 0:00.07 2 52 49 492K 5032K 2560K 144M
5364 SecurityAg 0.0% 0:00.01 5 112 166 2932K 15M 6632K 240M
5363 authorize 0.0% 0:00.02 1 31 30 580K 2556K 1564K 20M
4866 Motion 0.0% 0:09.00 29 214 791 21M 32M 62M 447M
4735 thunarInt 0.0% 0:00.00 1 0 25 24K 292K 136K 10M
4733 thunarInt 0.0% 0:00.00 1 0 24 0 292K 180K 10M
4732 thunarInt 0.0% 0:00.19 1 10 25 76K 292K 236K 10M
4730 thunarInt 0.0% 0:00.74 1 14 24 12K 292K 416K 10M
4780 xnet-gripd 0.0% 0:00.00 1 15 24 0 104K 156K 20M
4780 xnet-shop 0.0% 0:00.03 1 10 25 32K 148K 184K 22M
4782 xnet-net1 0.0% 0:00.00 1 0 24 0 104K 16K 20M
4780 xnet-net1 0.0% 0:00.00 1 0 24 0 104K 16K 20M
4690 xnet-shop 0.0% 0:00.03 1 10 26 56K 244K 200K 22M
4692 xnet-netd 0.0% 0:00.12 1 9 29 88K 104K 112K 20M
4527 AppleSpell 0.0% 0:00.03 1 22 31 409K 6156K 500K 34M
5359 xps10P105 0.0% 0:04.01 2 41 59 500K 112K 976K 2M
```

# 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.



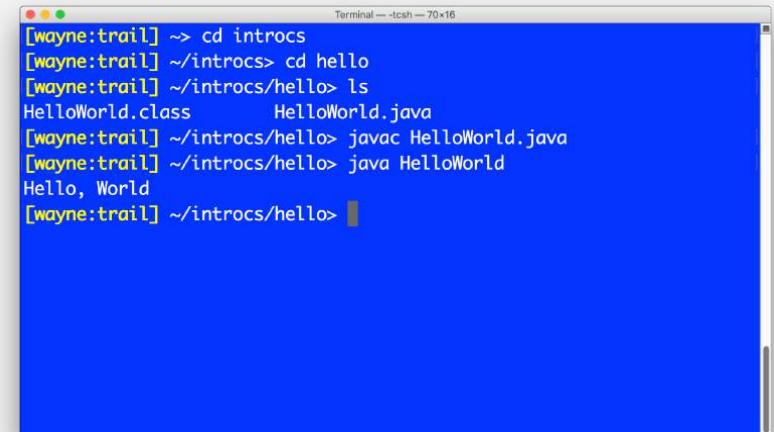
Bash coming to Windows



# Software for program development: tradeoffs

## Command line advantages.

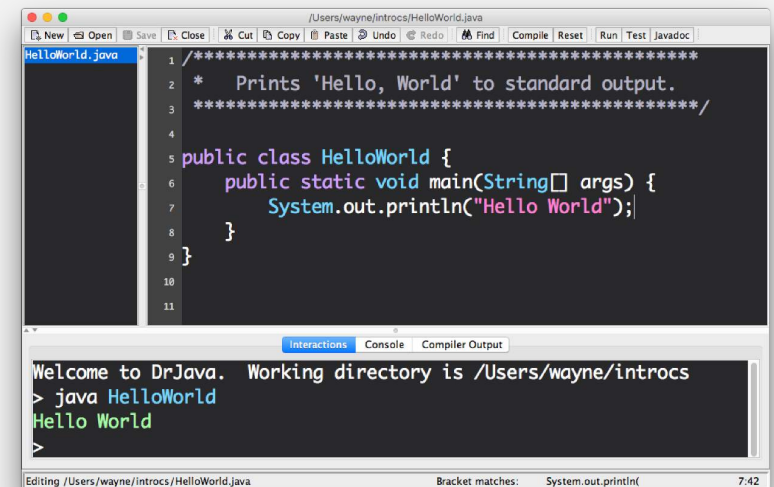
- More control over system.
- Approach works with any language.
- Easy to automate tasks via scripting.



```
Terminal -- -tosh -- 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.



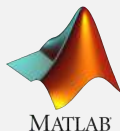
```

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

---

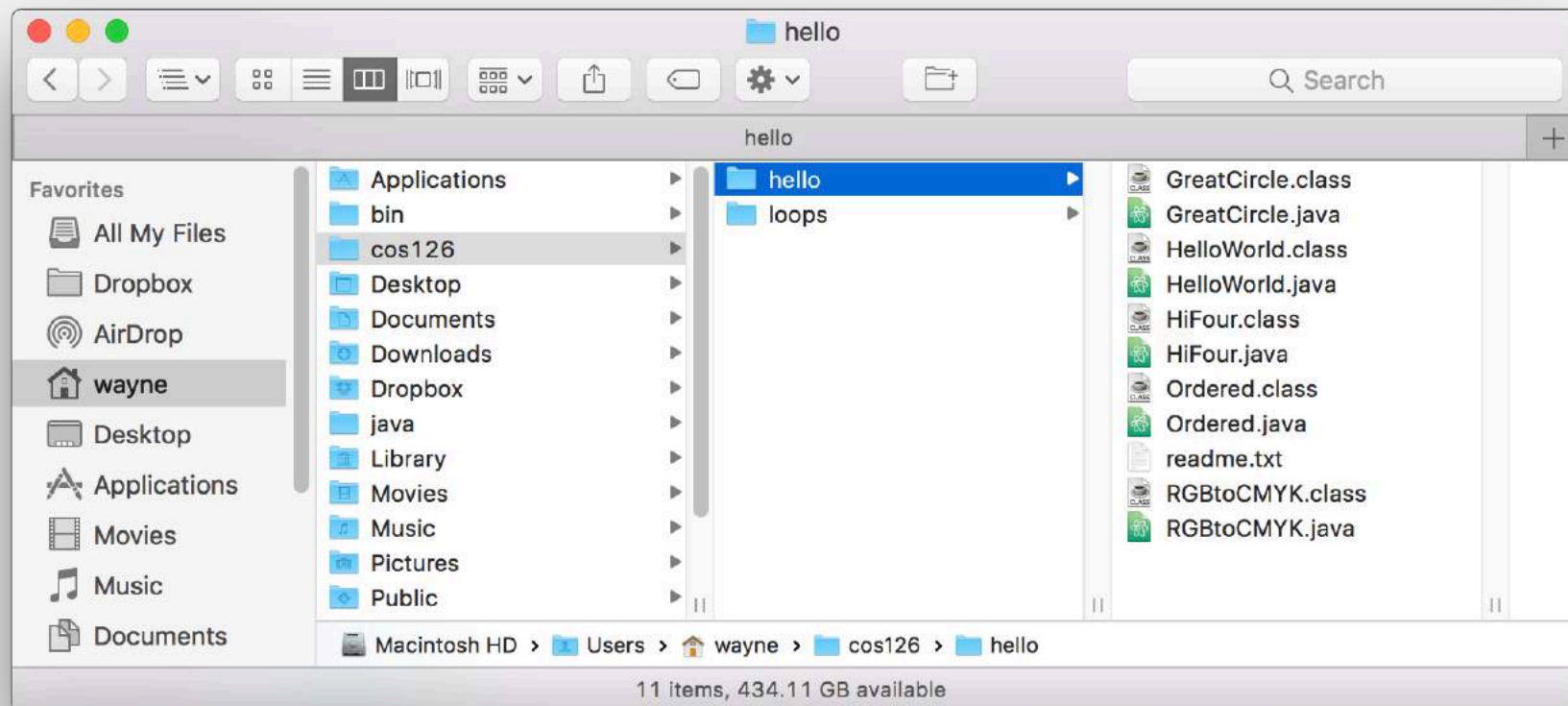
**File.** Sequence of bits stored on a computer.

**Filename.** String that uniquely identifies a file.

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

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

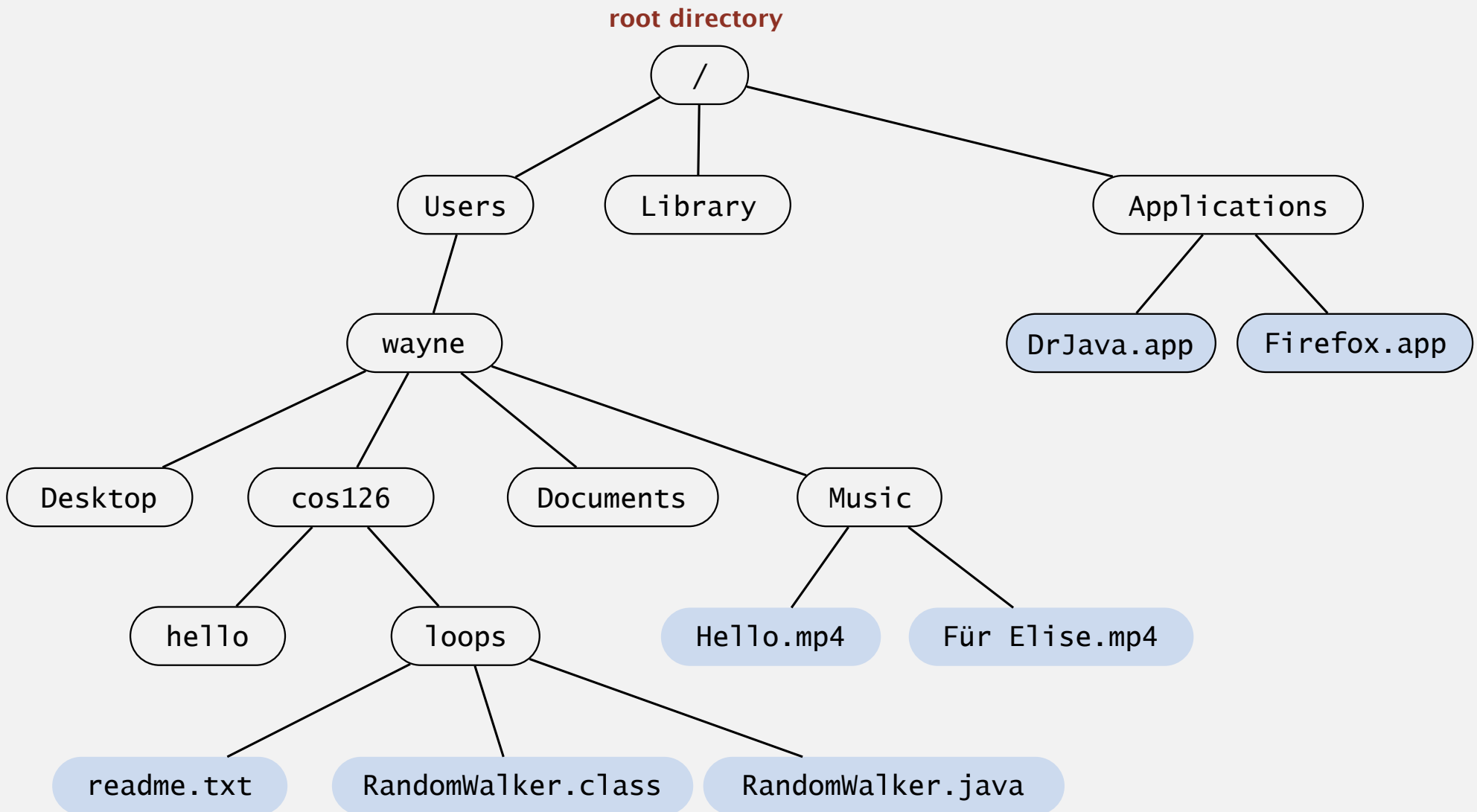
← note: forward vs.  
← backward slash



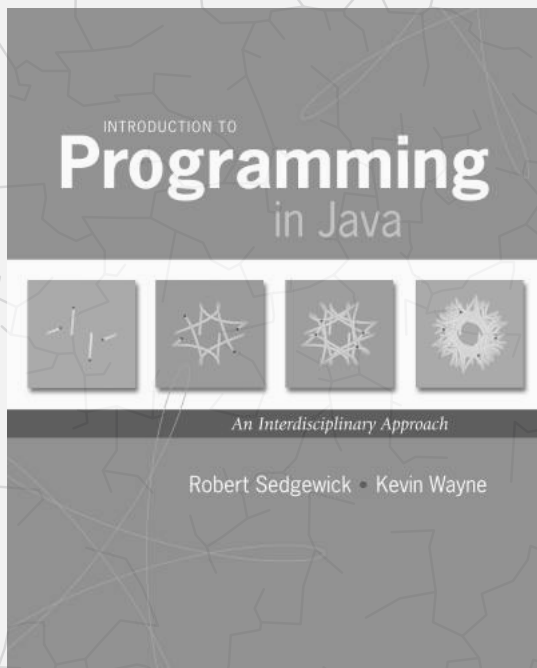
# File system hierarchy

---

**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*

# 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>&lt;Up Arrow&gt;</code> <code>&lt;Down Arrow&gt;</code>	
<i>autocomplete</i>	<code>javac Hello&lt;Tab&gt;</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>	

# Live demo of file system commands

---

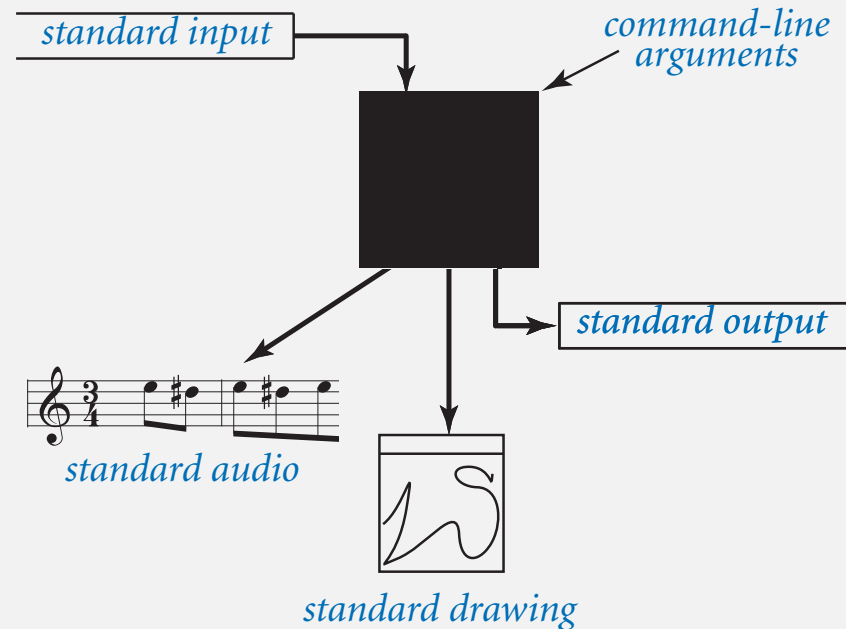
# Standard input and output abstractions

---

**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).





# 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>

# Live demo of standard input and output commands

---