Index of /~cs61b/lectures/02

Name	Last modified	Size	Description
Parent Directory	24-Sep-98 21:22	-	
CopyFile.java	26-Aug-98 09:15	1k	
SimpleIO.java	26-Aug-98 09:15	1k	
StringUse.java	26-Aug-98 09:15	1k	
WHWWW.java	26-Aug-98 09:15	1k	

```
                     CS61b: Lecture 2
                    Wednesday, August 26

REVIEW
------
If we have an object named x:
   x.methodName(...)    invokes a method, i.e., sends a message to x
   x.field              gets an object (or value) named field inside x

USING CLASSES AND OBJECTS:
--------------------------
Here's a simple Scheme program using strings:
   (let ((s1 "") (s2 "burger"))
      (set! s1 s2)
      (set! s1 "hamlet")
      (set! s1 (substring s1 0 3))
      (set! s1 (string-append s1 s2))
      (write s1)
   )

What are the steps in using a String in Java?

   1) declare a variable of type String (needs to give its type).
   2) initialize that variable to reference a new String object.
   3) call whatever methods we want on that object (by naming the variable)

String s1;                   // step 1
s1 = new String();           // step 2
String s2 = "special";       // steps 1 & 2, combined
s1 = s2;                     // assignment -- now they refer to the same object
s1 = s1.substring(0,4);      // now they refer to different objects
s1 = s1.concat("k");         // also written: s1 = s1 + "k";

I/O CLASSES AND OBJECTS IN JAVA: INPUT FROM THE KEYBOARD
--------------------------------------------------------
Here are some basic objects in the System for interacting with a user:

 System.out is a PrintStream object that outputs to the screen
 System.in is an InputStream object that reads from the keyboard

But System.in doesn't have methods to read a line directly. There
```

is a method called readLine that does, but it is defined on BufferedReader.

- How do we construct a BufferedReader? One way is with an InputStreamReader. (Look in the BufferedReader documentation. The constructors are also named BufferedReader. One takes an InputSteamReader argument.)
- How do we construct an InputStreamReader? We need a InputStream.
- How do we construct an InputSteam? System.in is one.

Aside: See the Java library API from the course home page for more I/O methods:
  lookup by "package," e.g., java.io
  lookup by method name, e.g., readLine, in the "index"

```
import java.io.*;
class SimpleIO {
  public static void main (String [] arg) throws Exception {
    BufferedReader keybd =
      new BufferedReader(new InputStreamReader(System.in));
    System.out.println(keybd.readLine());
  }
}
```

To use the Java libraries, other than java.lang, you need to "import" them.
To use most of the I/O methods, you need to add a "throws" to the function.

CLASSES FOR ACCESSING STUFF ON THE WEB

Let's say we want to read a line of text from the White House web page.
(Note: this will be HTML, which looks ugly, and you don't need to understand how to write it, even though it's in the book. Basically, it contains a lot of "<" and ">" characters.)

How to read a line of text? With ReadLine on BufferedReader.
How to create a BufferedReader? With an InputStreamReader.
How to create a InputStreamReader? With a BufferedInputStream.
How to get a BufferedInputStream? With a URLConnection (tricky API).
How to get a URLConnection? With a URL.

```
import java.net.*;
import java.io.*;

class WHWWW {
  public static void main(String[] arg) throws Exception {
    URL u = new URL("http://www.whitehouse.gov/");
    URLConnection uC = u.openConnection();
    BufferedInputStream ins = (BufferedInputStream) uC.getContent();
    InputStreamReader isr = new InputStreamReader(ins);
    BufferedReader whiteHouse = new BufferedReader(isr);
    System.out.println(whiteHouse.readLine());
  }
}
```

MORE I/O CLASSES: FILE INPUT AND OUTPUT

```
import java.io.*;

class CopyFile {
  public static void main(String arg[]) throws Exception {
    // Get both file names
    BufferedReader keybd =
      new BufferedReader(new InputStreamReader(System.in));
    System.out.print("name of file to copy: ");
```

```
    System.out.flush();
    String fileNameOrig = keybd.readLine();
    String fileNameCopy = fileNameOrig.concat(".copy");

    // Arrange to read from original file
    BufferedReader readerOrig =
      new BufferedReader(new InputStreamReader(
        new FileInputStream(new File(fileNameOrig))));

    // Arrange to write to backup file
    PrintWriter writerCopy = new PrintWriter(
      new FileOutputStream(
        new File(fileNameCopy)), true);

    // Read a line from original file and write to backup copy
    writerCopy.println(readerOrig.readLine());
    }
}
```

Notes:
1) the book uses an outdated (deprecated) class called PrintStream,
  rather than PrintWriter. You should not use such classes, or anything
  else that the compiler prints warnings about.
2) All code for this lecture is online in ~cs61b/public_html/lectures/02
  and on the web page.
3) Everything you need for the Java problem in hw1 is there.