Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-two)

2.8: Basic I/O in Java

Java supports a very rich and powerful I/O architecture, but to understand it you've got to get some of the concepts of object-oriented programming under your belt. We're still at that awkward stage, so for now, here are a few idioms you can just take for granted that will get you up and running.

System.out.print() and System.out.println() are the workhorse functions for displaying text on the console. They will accept any type of Java data and display it. println() appends a carriage return; print() does not. Given what you know about Strings, you can print just about anything using them:

```
System.out.println(3); // OK
System.out.println(3.2); // OK
System.out.println("The answer is " + 3.2); // OK
```

Sometimes after you print() something, nothing appears on the screen until you call System.out.flush(). println() doesn't have this "problem."

Understanding input is a bit tougher. You're going to have to take my word for it, but if you want to read lines from the console, you need to do something like this:

```
// put this at the very top of your file:
import java.io.*;

// change the first line of main to look like this
public static void main(String [] argv) throws IOException
{
// then, in your main() function:
BufferedReader stdin =
   new BufferedReader(new InputStreamReader(System.in), 1);

// then, to get a line of user input
//as a string:
String s = stdin.readLine();

// ... or to read an integer
int i = Integer.parseInt(stdin.readLine());
```