

Play with *DrJava* & the Scanner class

The programming environment that we are going to use is called *DrJava*. It provides both the abilities to execute a program and to work interactively with the system.

In Java, getting input from users requires the use of an *object*. We'll talk much more about objects and classes later, but for now you will just get some experience with using an object from the **Scanner** class to getting keyboard input.

Compiling and running a program

1. Download from the *Lecture Materials* page the file `Earth.java` under the Oct 13 lecture. Save it to a directory of your choice, e.g., the Desktop. Now start up *DrJava* and open the file `Earth.java`. Read the code to make sure you understand it. Have you noticed that you can use a `printf` method for printing? `printf` behaves just like `fprintf` from MATLAB. Compile the file (click *Compile All*) and then run it (*select Tools—Run Document's main Method*).
2. Change the print statement to use the `println` method. (You cannot control the number of decimal places here.) Compile and run the program again.
3. Now change the program to accept user input from the keyboard. Remove the statement that declares variable `r` and assigns a value to it. You will need an object from the **Scanner** class to read input. The following statement will create such an object and call it `keyboard`:

```
Scanner keyboard= new Scanner(System.in);
```

You need this object before reading input, so put this statement inside the `main` method at the top. Before we go further, compile the code and you'll see that there's an error! Read the error message—it says that the class **Scanner** is unknown. **Scanner** is not a standard class that is automatically available, so we have to *import* it. Write the following code at the top of the file *above* the class header:

```
import java.util.Scanner;
```

Now the code compiles successfully. To prompt the user for input and to read the input, type the following statements:

```
System.out.print("Enter the radius: ");  
double r= keyboard.nextDouble();
```

The method for reading a value of type `double` is `nextDouble()`. Note that this method only reads, it *does not* allow you to specify a prompt the way the MATLAB function does. That's why we needed to use the print statement before reading a value to specify the “prompt.” You can probably guess the rest of the method names for reading different types of primitive values: `nextInt()`, `nextChar()`, `nextBoolean()`, etc.

Using *DrJava*'s interaction pane

You can type individual statements and expressions in the *Interaction pane* for experimentation without writing an entire method in a class. For example type the following statements in the *Interaction pane*:

```
boolean b= 9<3; //b now gets the boolean value false  
System.out.println(b);
```

The value `false` should be printed.

Primitive types in Java

Compute the following expressions in DrJava, either in the interactions window, or included in a short program. *Think about what should be the output before pressing the <Enter> key! The idea is to analyze, not to just copy the output without thinking!* For every expression, write the result on the line to the right and if there is a question try to answer it.

```
-4 - -4 - -4           -----
6 / 4                  ----- // Why is it not 1.5?
5 % 3                  -----
-5 % 3                 -----
Integer.MIN_VALUE     -----
Integer.MIN_VALUE - 1 ----- // Why?
Integer.MAX_VALUE     -----
Integer.MAX_VALUE + 1 ----- // Why?
true && false         -----
3 < 5 || 5 < 3        -----
"It is " + true       ----- // Why is it not an error?
Math.min(2, Math.max(1, 3)) -----
Math.abs(-2)          -----
Math.ceil(-5.1)       -----
Math.floor(-5.1)      -----
```

Conditionals

This may look familiar...

An instrument in a chemistry lab has a precision of 0.001 mg/L. This instrument gives two readings in an experiment. Write a java program to determine whether the two numeric readings R1 and R2 are the same given the precision of the instrument. The program should display R1, R2, and one of three messages: "R1 is higher," "R2 is higher," or "Readings are the same." Use *nested ifs* and call the program `Reading.java`. Let the variable values R1 and R2 be user input or use `Math.random()` to generate some values.

Delete your files from the computer before you leave!