Introduction to Computers and Engineering Problem Solving
1.00 / 1.001 Fall 2005

Problem Set 1
Due: 11AM, Friday September 16, 2005

Java Introduction 1 [100 points]

Introduction

This problem set will ask you to complete a bunch of short exercises to help you get familiar with java.

You might need to use the following:

JOptionPane.showInputDialog()

This is a way to get input from the user. This is a method. Fill in the parentheses with a message to the user (this is called the argument). This method will return a `String`. Suppose, for example, your program has the line of code:

```
String name = JOptionPane.showInputDialog("What is
your name?");
```

In this case, the program will display a dialog box with the text, "What is your name?" in it and a place for the user of the program to type some text. When they type the text and click on the button in the dialog box to indicate that they want to submit that text, the Java variable `name` will contain whatever `String` of text the user inputs.

Integer.parseInt()

This method takes a `String` as an argument and converts the `String` to an `int`. For example, if the variable `radius` is of type `String`, the line of code:

```
int r = Integer.parseInt(radius);
```

results in the variable `r` being assigned the `int` value corresponding to the `String` stored in `radius`.

Double.parseDouble()

This method takes a `String` as an argument and converts the `String` to a `double`. For example, the line of code:

```
double d = Double.parseDouble("3.14159");
```

Results in the value of `d` being set to 3.14159. Note that the variable `d` is a numerical value, while "3.14159" is a value of type `String`.

Math.sqrt()

This method takes a `double` and returns the square root of the argument as a `double`. For example, if the variable named `val` is a `double` with the value 4.0, the line of code:

```
double rootVal = Math.sqrt(val);
```

results in `rootVal` being assigned the value 2.0.

Math.round()

This takes a `float` as its argument and returns the nearest `int value`.

You should take a look at the javadocs for more information about any of these methods. http://java.sun.com/j2se/1.5.0/docs/api/

Assignment

Complete the following exercises by making them successive calculations in a single `main()` method of a single class:

`Introduction`

- Have your program ask the user his/her name and print out a message welcoming him/her.

`Cube`

- Have your program ask the user for the integer length of a side of a cube and print out its volume.

`Circle`

- Have your program ask the user for the real-valued radius of a circle and print out its area. Check the `Math` class for any constants you need.

`Equilateral triangle`

- Have your program ask the user for the integer length of a side of an equilateral triangle and print out its area as a double value. Make sure to use the following formula: $(1/4)(s^2)(\sqrt{3})$

`Trapezoid`

- Have your program ask the user the height of a trapezoid, as well as the 2 bases. Print out its area. All three inputs as well as the output value should be treated as double

`Slope of a Line`

- Have your program ask the user for 2 points, taking the integer valued x and y coordinate of each point separately. Have your program check if the points are the same and if they are, print out a message informing the user of this. Have your program check if the points form a vertical line and if they do, print out a message saying that the slope is undefined. If the slope is defined, have your program print out the slope as a double value.

`Time Conversion`

- Have your program ask the user for a length of time in seconds. Have your program print out the length of time in integer units of hours, minutes, and seconds.

Turn In

- Turn in **electronic** copies of **all source code** in a single .java file No printed copies are required.
- Place a **comment** with your full name, MIT server username, tutorial section, TA's name, and assignment number at the beginning of your `.java` file solution.
- Remember to **comment your code**. Points will be taken off for insufficient comments.
- Your solution is **due at 11AM**. Your uploaded file should have a time stamp of no later than 11AM on the due date.
- **Do not** turn in compiled byte code (`.class` files) or backup source code (`.java~` files).

Penalties

- **30% off** If you turn in your problem set **after 11AM on Friday** but **before 11AM on the following Monday**.
- **No Credit** If you turn in your problem set **after 11AM on the following Monday**.