CS 2334: Lab 1

General Procedures

- Most labs will have a short lecture component
  - May “follow along” with Eclipse
  - You are welcome to “run ahead” on the lab itself, but please do not disrupt the lecture
- The lecture will be followed by time to work on the assigned lab and interact one-on-one with the TAs
- The labs are designed to be completed by most in the lab session
- If you finish early and the TA has a chance to do some preliminary testing, then you may leave the session early

Download and Install

- Java JDK:

  http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

- Eclipse:

  http://www.eclipse.org/downloads

Task 1

- Create a new project called *Lab1*
- Create a class called *Driver*
  - The main method prints “Hello, World!”
- Compile and execute

(demo: Eclipse)

Task 2

Create a new class: *Forecast*

- One property, called *info* that is an array of 4 Strings

- One constructor method:

```
public Forecast(String strg)
```

where strg is “<type>,<low>,<high>”
<type> is a string
<low> and <high> are strings encoding positive integers

Task 2

Create a new class: *Forecast*

```
public Forecast(String strg)
```

- The constructor method splits the string into three parts and fills in the *info* property:
  - info[0] is the <type> in all upper case
  - info[1] is the <low> temperature
  - info[2] is the <high> temperature
  - info[3] is the difference between <high> and <low>

(demo: Java API: Strings)

Task 2

Forecast:

- toString() method:

```
public String toString()
```

- Describe the Forecast object
  - Constructor string: “sunny,29,53”
  - toString() output:

```
CONDITIONS: SUNNY, Low: 29, High: 53, Range: 24
```

Project-Level Documentation

Include at the top of every file:

```
/**
     @author: <your name>
     Date: <>
     Project: <number>
     <A short, abstract description of the file>
*/
```

Method-Level Documentation

How should we document the following method?

```
public static boolean isInRange(double min, double max, double value)
```

Method-Level Documentation

```
/**
	Indicate whether a value is within a range of values

	@param min Minimum value in the range
	@param max Maximum value in the range
	@param value The value being tested
	@return True if value is between min and max.  False if outside this
range.
*/

public static boolean isInRange(double min, double max, double value)
```

Inline Documentation

- Include inline documentation for each line of code, or small group of lines
- Capture the *logic* of what is happening (and why) – don't repeat what the code says

Inline Documentation

```
public static boolean isInRange(double min, double max, double value)
{
	// Check lower bound
	if(value < min)
		return false;

	// Check upper bound
	if(value > max)
		return false;

	// Within the boundaries
	return true;
}
```

Back to Task 2

Modify your Driver class main function:

```
public static void main(String [ ] args) throws IOException {
      BufferedReader br = new BufferedReader(
                              new InputStreamReader(System.in));
      String input = br.readLine();
```

- Use *input* to create an instance of Forecast
- Print Forecast instance

Testing
(some examples)

Input in console: ”Sunny,29,53”

Output:

CONDITIONS : SUNNY, Low: 29 , High: 53 , Range: 24

Input: “Rainy,38,42”

Output:

CONDITIONS : RAINY, Low: 38 , High: 42 , Range: 4

Generate Documentation with Javadoc

(demonstration)

Notes:

- Use *private* visibility
- Use the default destination
- You should check your documentation to make sure everything was included.

Exporting

(demonstration)

Notes:

- Export to zip file
- Include both the *src* and *doc* directories

Submission

- Due date: Friday, August 28th @11:59pm
- Use the lab1 dropbox on D2L