UNBC Laboratory Assignment
Computer Science 100—Fall 2012

Simple Input and Output in Java

Purpose:

To demonstrate an understanding of basic input and output in Java.

Due Date:

The completed lab assignment is due Friday, 21 September *at the beginning of lecture*.

Please hand in complete listings for all of your programs.

`script` and `scriptfix`

Once you have compiled your program and have it running, your lab instructor will show you how to use

- the `script` to record exactly what you have done,
- the `scriptfix` command to clean up your script files, and
- the "`enscript -2rG`" command to print your script file.

Hand in printed script files when you have completed a lab. Please hi-light your name (and *only* your name) before you hand it in.

`javadoc` and Documentation

Java programs can contain special comments that start with `/**` and end with `*/`. These comments are recognized by the `javadoc` program and can be used to create Html documentation automatically.

```
Celsius temperature to convert? 22.0
22.0 C is exactly 71.6 C.
22.0 C is approximately 74.00 C.
```

Figure 1: Sample I/O for Celsius to Fahrenheit program.

Until otherwise specified in a lab handout, put a comment that looks like

```
/**
 * This test8 demonstrates simple IO in Java.
 *
 * @author David Casperson 78303-0901
 * @version 2001-09-11
 */
```

right above the

```
public class test8
{
  public static void main(String [] args)
```

lines in each of your programs.

Elsewhere in your programs use //-comments freely to make notes to yourself or to explain tricky code to the lab assistant.

If you work with someone on a laboratory assignment, you **must** *put a //-comment to this effect in your source code.*

Converting Celsius to Fahrenheit

Temperatures measured in degrees Celsius can be converted to temperatures in degrees Fahrenheit using the formula:

$$F = \frac{9}{5}C + 32. \tag{1}$$

A simpler approximation that works well for near-room temperatures is

$$F = 2C + 30. \tag{2}$$

Write a program that produces output like that shown in Figure 1.

Computing your age in Giga-seconds

```
Your age in years is: 47.
Your age is 1.48 Gs.
```

Figure 2: Sample I/O for GigaSecond program.

```
Total seconds to convert? 7384
7384 seconds is 2 hours 3 minutes and 4 seconds.
```

Figure 3: Sample I/O for Hours, Minutes, Seconds program.

Derive the equations necessary[1], and write a program that asks you for your age in years, and then prints your age in gigaseconds. The input and output should look something like that shown in Figure 2.

Note that if you print your output using a command like

```
System.out.println("Your age is " + ageInGigaSeconds + " Gs.");
```

you are likely to get an output that looks like

```
Your age is 1.48318364736 Gs.
```

To get a nicer format, you can use a `DecimalFormat` object. To do this

1. Import `java.text.DecimalFormat`.
2. Create a `DecimalFormat` variable with a line like

   ```
   DecimalFormat gsFormat = new DecimalFormat(" 0.00 Gs");
   ```

3. Use the `gsFormat` object's to convert `doubles` to nice `Strings`. For instance

   ```
   "Fred "+gsFormat.format(2*0.74)+".Gertrude"
   ```

   should yield `"Fred 1.48 Gs.Gertrude"`. Adjust to taste.

Converting seconds into hours minutes and seconds

There 60 seconds in a minute, and 60 minutes in an hour. Work out equations to convert total seconds into hours, minutes and seconds using integer divide and '%'. Write a program that produces output[2] like that shown in Figure 3.

[1] Assume either 365 days in a year, or 365.2442 days in a year. In other words, you *do not* need to calculate exactly when there were leap-years. You *should* put comments in your program explaining what you chose to do.
1 $Gs = 10^9\ s$. In Java, 10^9 written as a `double` is `1.0E9`.

[2] For this lab it is ok to produce output like "**3661 seconds is 1 hours 1 minutes and 1 seconds.**" If you have previous programming experience, see if you can get the plurals right.