

## Static Methods and Constants

---

### Purpose:

To understand the use of static methods.

---

### Due Date:

The completed lab assignment is due Friday 2011-10-07 *at the beginning of lecture*.

---

### Computing with static methods and constants

Redo the *first two* programming questions from Laboratory Assignment 2 using at least the following methods. For each assignment use:

```
public static double promptForDouble(String prompt) {...}
```

For the Fahrenheit to Celsius conversion use methods like:

- `public static double cToF_exact(double f) {...}`
- `public static double cToF_approx(double f) {...}`

(You may change the names to be more descriptive.) For the year to gigasecond and gigasecond to year questions, use `public static final double` constants

- `SECONDS_PER_YEAR`
- `SECONDS_PER_GIGASECOND`

and methods

- `public static double yearsToGs(double y) {...}`

*You must use methods (other than `public static void main(Strings args[])`) in order to get credit for this part.*

⇒ Hand in the redone assignment 2 along with the solution to the following problem. Be sure to provide a JAVADOC-style comment before each method.

---

## Calculate the number of days that you have been alive

The problem here is to compute the number of days that you have been alive (on the day that the program is run). It is possible to do this from scratch using the `java.util.GregorianCalendar` class, but correct calculations involve subtleties that are not the point of this lab assignment.

For this reason, you may use the supplied code for the methods

---

`daysSince1Jan1970`

`(year, month, day)` which returns the number of whole days between January 1, 1970, and the argument. For instance `daysSince1Jan1970(1970, 1, 1)` returns 0. Note that unlike the JAVA library methods, the months are numbered from 1.

`getCurrentYear()` returns the current year (2011).

`getCurrentMonth()` returns the current month (10). Note that unlike the JAVA library methods, the months are numbered from 1.

`getCurrentDay()` returns the current between (between 1 and 31 inclusive).

---

Do you see how to use the `daysSince1Jan1970` method to compute the number of days between *any* two dates? Write a method

```
public static int daysBetween(int year1, int month1, int day1,
                              int year2, int month2, int day2)
{...}
```

that computes the number of days between two dates. Then write a function that asks the user for her birth year, birth month (as number from 1 to 12), and birth day (of the month), then prints out the number of days that she has been alive. Sample output is shown in Figure 1 on page 5.

How can you test it?

⇒ Script your code, your compilation, and some test runs. Random people may accidentally see your script file, so use a fictitious birth date if you wish to keep your own birth date private. Be sure to use `/opt/scriptfix/scriptfix` on script.

## Laboratory Assignment 2 problems

The following problem descriptions are copied from Laboratory Assignment 2 for your convenience. See the section “Computing with static methods” on the first page for what to do with them.

### Converting Celsius to Fahrenheit

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

$$F = \frac{9}{5}C + 32. \quad (1)$$

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

$$F = 2C + 30. \quad (2)$$

Write a program that produces output like that shown in Figure 2 on page 5.

### Computing your age in Giga-seconds

Derive the equations necessary<sup>1</sup>, 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 3 on page 5.

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 need to use a `DecimalFormat` object. To do this

1. Put the line `import java.text.DecimalFormat ;` at the very top of your program.
2. Create a `DecimalFormat` variable with a line like

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

3. Print your output with a line like

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

<sup>1</sup>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. In JAVA,  $10^9$  written as a `double` is `1.0E9`.

Figure 1: Sample I/O for the Days Alive program.<sup>‡</sup>

---

```
Year of birth? 1959
Month of birth (1 for Jan)? 4
Day of month of birth? 30
You have been alive for 19148 days.
```

---

<sup>‡</sup>This calculation is for Prime Minister Stephen Harper. The calculation was done on Sunday, 2011-10-02. If you are politically curious, determine how many days Jack Layton lived.

Figure 2: Sample I/O for Celsius to Fahrenheit program.

---

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

---

Figure 3: Sample I/O for GigaSecond program.

---

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

---