UNBC Laboratory Assignment
Computer Science 100—Winter 2008

Static Methods and Constants

Purpose:

To understand the use of static methods to decompose problems.

To demonstrate familiarity with `%` and integer-`/` operators by computing the date of Easter.

Due Date:

The completed lab assignment is due Wednesday 2008-02-06 *at the beginning of lecture*.

Using `static` methods and constants to compute

By the end of the Monday, 2008-01-28 lecture you should be able to write `static` methods to compute mathematical functions. Redo the programming questions from Laboratory Assignment 2 using at least the following functions. For each assignment use:

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

For the Fahrenheit to Celsius conversion use functions like:

- `public static double fToC_exact(double f) {...}`
- `public static double fToC_approx(double f) {...}`

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

- `SECONDS_PER_YEAR`
- `SECONDS_PER_GIGASECOND`

and methods

- `public static double yearToGs(double y) {...}`
- `public static double GsToYears(double gs) {...}`

⇒ 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.

Computing the date of Easter

An extreme demonstration of the utility of integer division and modulus operations comes

```
a = y %  19 = 2008% 19  = 13;
b = y / 100 = 2008 / 100 = 20;
c = y % 100 = 2008 / 100 = 8 ;
d = b /   4 = 20 /   4 = 5 ;
e = b %   4 = 20 / 4 = 0 ;
g = (8*b+13) / 25 = (8*20+13) / 25 = 6 ;
h = (19*a + b - d -g +15 ) % 30 = (19*13 + 20 - 5 - 6 +15 ) % 30 = 1 ;
n = (h - m + r + 90) / 25 = (1 - 0 + 0 + 90) / 25 = 3 ;
p = (h - m + r + n + 19) % 32 = (1 - 0 + 0 + 3 + 19) % 32 = 23 ;
```

Figure 1: Sample calculation of the date of Easter Sunday

from implementing the following algorithm to compute the date of Easter.[1] The algorithm was invented by Carl Friedrich Gauss in 1800.

Gauss' Algorithm for Easter

1. Let `y` be the year (such as 1800 or 2001).
2. Divide `y` by 19 and call the remainder `a`. Ignore the quotient.
3. Divide `y` by 100 to get a quotient `b` and a remainder `c`.
4. Divide `b` by 4 to get a quotient `d` and a remainder `e`.
5. Divide `8*b + 13` by 25 to get a quotient `g`. Ignore the remainder.
6. Divide `19*a + b - d - g + 15` by 30 to get remainder `h`. Ignore the quotient.
7. Divide `c` by 4 to get a quotient `j` and a remainder `k`.
8. Divide `a + 11*h` by 319 to get a quotient `m`. Ignore the remainder.
9. Divide `2*e + 2*j - k - h + m +32` by 7 to get remainder `r`. Ignore the quotient.
10. Divide `h - m + r + 90` by 25 to get a quotient `n`. Ignore the remainder.
11. Divide `h - m + r + n + 19` by 32 to get a remainder `p`. Ignore the quotient.

Then Easter falls on day `p` of month `n`.

end Gauss' Algorithm for Easter

Figure 1 shows the computation for this year.

Write a Java program to perform this computation. Have it ask the user for a year, and then print out the date as, for instance, `"Easter Sunday 2008 falls on March 23"`.

Have your program check the year for reasonableness. This is notoriously difficult as different countries switched to the Gregorian calendar in different years. For the sake of this problem, assume that the year is reasonable if it falls between 1750 and 2400.

Try to use `static` methods to make the computations easier. Here are some suggested method signatures (remember to add `public static` and the method body):

- `int promptForInt(String prompt)`

[1] This problem comes from *pp.* 130–131 *Big Java* by Cay Horstmann published by Wiley (2002).
More precisely this calculation of the date of Easter according to the Gregorian calendar, which is the calendar currently in use in America and most of Europe.

```
Fahrenheit temperature to convert? 72.5
72.5 F is exactly 22.5 C.
72.5 F is approximately 21.25 C.
```

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

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

Figure 3: Sample I/O for GigaSecond program.

- `boolean yearIsReasonable(int year)`
- `String monthName(int month)`

Add any others that you find helpful.

Laboratory Assignment 2 problems

The following problem descriptions are copied from Laboratory Assignment 2 for your convenience. See the section "Using `static` methods to compute" on the first page for what to do with them.

Converting Fahrenheit to Celsius

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

$$C = \frac{5(F-32)}{9}. \tag{1}$$

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

$$C = (F-30)/2. \tag{2}$$

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

Computing your age in Giga-seconds

Derive the equations necessary, 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.

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));
   ```

Computing Walter Taylor's date of birth.

Write a program to solve the following problem.

Dr Walter Taylor (`http://genealogy.math.ndsu.nodak.edu/id.php?id=28490&fChrono=1`) had a conference held in his honor in August 2004. At the time of the conference Dr Taylor was nearly 2 Gs old. In what year was Dr Taylor born?