

CSE 142, Winter 2007
Programming Assignment #1: Six Days of "Xmas" (10 points)
Due: Thursday, January 11, 2007, 9:00 PM

Program Description:

This program tests your understanding of using static methods and `println` statements. You should write a Java class called `SixDays` that should be saved into a file called `SixDays.java`. Your program should produce the following song as output:

```
On the 1st day of "Xmas", my true love sent to me  
a partridge in a pear tree.
```

```
On the 2nd day of "Xmas", my true love sent to me  
two turtle doves, and  
a partridge in a pear tree.
```

```
On the 3rd day of "Xmas", my true love sent to me  
three French hens,  
two turtle doves, and  
a partridge in a pear tree.
```

```
On the 4th day of "Xmas", my true love sent to me  
four calling birds,  
three French hens,  
two turtle doves, and  
a partridge in a pear tree.
```

```
On the 5th day of "Xmas", my true love sent to me  
five golden rings,  
four calling birds,  
three French hens,  
two turtle doves, and  
a partridge in a pear tree.
```

```
On the 6th day of "Xmas", my true love sent to me  
six geese a-laying,  
five golden rings,  
four calling birds,  
three French hens,  
two turtle doves, and  
a partridge in a pear tree.
```

The song is a modified version of a classic holiday song. For brevity, we reduced the number of days from the original twelve to six and shortened Christmas to "Xmas".

You should **exactly** reproduce the format of this output. This includes having identical wording, spelling, spacing, punctuation, and capitalization. Please do not include additional verses, such as writing twelve days to match the complete song. You may include a blank line at the very end of the output if you like.

One way to write this program would be to simply write a `println` statement that outputs each line of the song in order. However, such a solution would not receive full credit. Part of the challenge of this assignment lies in recognizing the structure and redundancy of the song and improving the code using static methods.

Stylistic Guidelines:

You should not place any non-blank `println` statements in your `main` method. Instead, use static methods in this program, for two reasons:

1. To capture the *structure* of the song's six verses.

You should be using static methods to capture the structure of the song. You should, for example, have a method for each of the six verses of the song to print that verse's entire contents.

2. To avoid simple *redundancy* in the output.

You should use only one `println` statement for each distinct non-blank line of the song. For example, the following line appears several times in the output, but you should have only one `println` statement in your program that prints that line of the song:

```
a partridge in a pear tree.
```

However, a method that prints a single line such as the above is not useful. Instead, you should identify groups of two or more lines that appear in multiple places in the song and create static methods that capture those groups and are called multiple times. There is a general structural redundancy to the song that you should eliminate with your static methods. Recall that methods can call other methods if necessary. The key question to ask yourself is whether or not you have repeated lines of code that could be eliminated if you structured your static methods differently. As a point of reference, our solution to this program has twelve static methods other than `main` and occupies 89 lines including comments and blank lines.

You do NOT have to eliminate redundancy in lines that are similar but not identical, such as these:

```
On the 1st day of "Xmas", my true love sent to me  
On the 2nd day of "Xmas", my true love sent to me
```

It is not possible to avoid this partial-line redundancy using just what we have learned so far (static methods and simple `println` statements), so you are not expected to eliminate it.

Include a comment at the beginning of your program with some basic information and a description of the program. The comments in your program should be written in your own words and not copied from this document. For example:

```
// Suzy Student  
// CSE 142, Autumn 2049, Section XX  
// Programming Assignment #1, 06/07/49  
//  
// This program's behavior is ...
```

For this assignment, you should limit yourself to the Java features covered in Chapter 1 of the textbook. Though we will cover Chapter 2 while you work on this assignment, please do not use Chapter 2 features such as mathematical expressions, `print` statements (as opposed to `println`), or `for` loops on this program.

Submission and Grading:

Turn in your `SixDays.java` file electronically from the Assignments link on the course web page. Please make sure to use **exactly** this file name, including identical capitalization.

Part of your program's score will come from its "external correctness." External correctness measures whether the output matches **exactly** what is expected. (We are *very picky* about the output matching exactly. Every character and space must match.) Programs that do not compile will receive no external correctness points.

The rest of your program's score will come from its "internal correctness." Internal correctness measures whether your source code follows the stylistic guidelines specified in this document. This includes having an adequate comment header and capturing the structure and redundancy of the song as specified previously.