

## Programming Project #1: Song (20 points)

Leveraged from Building Java Programs: A Back to Basics Approach Second Edition

By Stuart Reges and Marty Stepp

**Due by Monday September 15, 2014 beginning of class**

### Program Description:

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

#### Song Option 1:

There was an old woman who swallowed a fly.  
I don't know why she swallowed that fly,  
Perhaps she'll die.

There was an old woman who swallowed a spider,  
That wriggled and iggled and jiggled inside her.  
She swallowed the spider to catch the fly,  
I don't know why she swallowed that fly,  
Perhaps she'll die.

There was an old woman who swallowed a bird,  
How absurd to swallow a bird.  
She swallowed the bird to catch the spider,  
She swallowed the spider to catch the fly,  
I don't know why she swallowed that fly,  
Perhaps she'll die.

There was an old woman who swallowed a cat,  
Imagine that to swallow a cat.  
She swallowed the cat to catch the bird,  
She swallowed the bird to catch the spider,  
She swallowed the spider to catch the fly,  
I don't know why she swallowed that fly,  
Perhaps she'll die.

There was an old woman who swallowed a dog,  
What a hog to swallow a dog.  
She swallowed the dog to catch the cat,  
She swallowed the cat to catch the bird,  
She swallowed the bird to catch the spider,  
She swallowed the spider to catch the fly,  
I don't know why she swallowed that fly,  
Perhaps she'll die.

There was an old woman who swallowed a horse,  
She died of course.

#### Song Option 2:

Bought me a cat and the cat pleased me,  
I fed my cat under yonder tree.  
Cat goes fiddle-i-fee.

Bought me a hen and the hen pleased me,  
I fed my hen under yonder tree.  
Hen goes chimmy-chuck, chimmy-chuck,  
Cat goes fiddle-i-fee.

```
Bought me a duck and the duck pleased me,  
I fed my duck under yonder tree.  
Duck goes quack, quack,  
Hen goes chimmy-chuck, chimmy-chuck,  
Cat goes fiddle-i-fee.
```

```
Bought me a goose and the goose pleased me  
I fed my goose under yonder tree.  
Goose goes hissy, hissy,  
Duck goes quack, quack,  
Hen goes chimmy-chuck, chimmy-chuck,  
Cat goes fiddle-i-fee.
```

```
Bought me a sheep and the sheep pleased me,  
I fed my sheep under yonder tree.  
Sheep goes baa, baa,  
Goose goes hissy, hissy,  
Duck goes quack, quack,  
Hen goes chimmy-chuck, chimmy-chuck,  
Cat goes fiddle-i-fee.
```

```
Bought me a pig and the pig pleased me,  
I fed my pig under yonder tree.  
Pig goes oink, oink,  
Sheep goes baa, baa,  
Goose goes hissy, hissy,  
Duck goes quack, quack,  
Hen goes chimmy-chuck, chimmy-chuck,  
Cat goes fiddle-i-fee.
```

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. 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 `println` statements in your `main` method. (It is okay for `main` to have empty `println` statements to print blank lines.) Instead of printing in `main`, use static methods for two reasons:

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

You should write static methods to capture the structure of the song. You should, for example, have a method for each of the 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:

```
Perhaps she'll die.
```

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.

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

```
There was an old woman who swallowed a spider,  
There was an old woman who swallowed a bird,
```

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. See the example below. Also include a comment on a line right before each method. The comments in your program should be written in your own words and not copied from this document.

```
/**  
 * Programming Project #1, Song  
 * This program's behavior is . . .  
 *  
 * by Sarah Student  
 * AP CS, 2014-2015  
 * September 15, 2014  
 */
```

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

## Submission and Grading:

Turn in your **lastnamesong.java** file electronically in Edmodo by **Monday September 15, 2014 beginning of class**. Please make sure to use this file name structure. Part of your program's score will come from its "external correctness." External correctness measures whether the output matches exactly what is expected. 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 adequate class and method comments and capturing the structure and redundancy of the song as specified previously.