CSE 142: Computer Programming I   Summer 2020

Assignment 1: Song (10 points)   *due June 30, 2020, 11:59pm*

This assignment focuses on static methods and `System.out.println` statements. Turn in the following file using the link on the course website:

- `Song.java` – A program that produces the output described below

Program Behavior

You will write a Java program that produces as output a cumulative song in which successive verses build on previous verses (as described in http://en.wikipedia.org/wiki/Cumulative_song). In particular, we will produce the song "There Was an Old Woman Who Swallowed a Fly." Your program should produce **exactly** the output below, except for the custom verse (see below). You can use the Output Comparison Tool on the course web site to check if your output is correct.

You must **exactly** match the output here, including both content and format. Check your output carefully!

```
┌─ Expected Output ─┐
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.

<< Your custom sixth verse goes here >>

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

Custom Verse

As indicated above, you should include a custom sixth verse that matches the pattern of the first five verses. This custom verse should be printed in place of "« *Your custom sixth vese goes here* »" in the output above. (You should *not* print that placeholder text.) For example, some versions of the song have a sixth verse for swallowing a goat ("Just opened her throat to swallow a goat"). Notice that the first two lines should either end in the same word (fly/fly, bird/bird, cat/cat, etc.) or should end with rhyming words (spider/inside her). You should not simply copy one of the previous animals or to use the verses you'll find on the web (e.g., goat and cow); you should write your own custom verse. The text of the verse should not include hateful, offensive, or otherwise inappropriate speech.

Style Guidelines

You should follow all guidelines in the Style Guide and on the General Style Deductions page of the course website. Pay particular attention to the following elements:

Capturing Structure

You should use static methods to accurately capture the structure of the song in your code. You must, for example, have a separate method for each of the seven verses of the song (verses are separated by blank lines in the output). As a result, you will not have any `println` statements in main except perhaps a `println` that produces a blank line.

Avoiding Redundancy

You should also use static methods to avoid "full-line" redundancy. In particular, you **must** make sure that you use only one `println` statement for each distinct line of the song. For example, the line:

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

appears several times in the output. To receive full credit, you must have only one `println` statement in your program that produces this line.

On the other hand, you are *not* required to fix "partial-line" redundancy, as in pairs of lines like these:

```
    There was an old woman who swallowed a horse,
    There was an old woman who swallowed a dog,
```

or these:

```
    She swallowed the dog to catch the cat,
    She swallowed the dog to catch the bird,
```

It is not possible to avoid this type of redundancy using material we have covered so far (methods and `println` statements), so *you are not expected to do so.*

Permitted Java Features

You are not allowed to use more advanced features than what we have covered in class. For this assignment, you should limit yourself to the Java features covered in chapter 1 of the textbook. You should not use `System.out.print()` statements even though they are covered in chapter 1. All output (*including the last line*) should be produced using `System.out.println()`. You also may not use the `\n` escape sequence.

Indentation and Whitespace

Your program should be properly indented and make proper use of blank lines as shown in class and discussed in the Style Guide. The Indenter Tool on the course website can help you with indentation, but you should still check the indentation yourself.

Header Comment

You should include a comment at the beginning of your program with some basic identifying information and a description of the program. Your comment should look something like this:

```
// Grace Hopper          << replace with your name >>
// 01/14/2020
// CSE142
// TA:  Ada Lovelace      << replace with your TA's name >>
// Assignment #1
//
// This program will...   << add a brief description of the program >>
```

You can include additional information if you like, but make sure at least these details are present. Your program description should be specific to this assignment, but not include any implementation details. See the Style Guide for more information.

Getting Help

If you find you are struggling with this assignment, make use of all the course resources that are available to you, such as:

- Reviewing relevent lecture examples

- Reviewing this week's section handouts

- Reading the textbook

- Visiting the IPL

- Posting a question on the message board

Academic Integrity

Remember that, while you are encouraged to use all resources at your disposal, including your classmates, **all work you submit must be entirely your own**. In particular, you should **NEVER** look at a solution to this assignment from another source (a classmate, a former student, an online repository, etc.). Please review the full policy in the syllabus for more details, and ask the course staff if you are unclear on whether or not a resource is OK to use.