CPW 142 Object-Oriented Programming I, Summer 2016
Programming Assignment #1

Your first program will require the use of static methods and **`println`** statements.
You are going to write a Java program that produces as output the following song.

```
There was an old lady who swallowed a fly.
I dunno why she swallowed that fly,
Perhaps she'll die.

There was an old lady who swallowed a spider,
That wiggled and wiggled and tickled inside her.
She swallowed the spider to catch the fly.
I dunno why she swallowed that fly,
Perhaps she'll die.

There was an old lady who swallowed a bird;
How absurd, to swallow a bird!
She swallowed the bird to catch the spider
That wiggled and wiggled and tickled inside her.
She swallowed the spider to catch the fly.
I dunno why she swallowed that fly,
Perhaps she'll die.

There was an old lady who swallowed a cat.
Imagine that, she swallowed a cat.
She swallowed the cat to catch the bird ...
She swallowed the bird to catch the spider
That wiggled and wiggled and tickled inside her.
She swallowed the spider to catch the fly.
I dunno why she swallowed that fly,
Perhaps she'll die.

There was an old lady 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
That wiggled and wiggled and tickled inside her.
She swallowed the spider to catch the fly.
I dunno why she swallowed that fly,
Perhaps she'll die.

There was an old lady who swallowed a goat.
Just opened her throat and swallowed a goat!
She swallowed the goat to catch the 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
That wiggled and wiggled and tickled inside her.
She swallowed the spider to catch the fly.
I dunno why she swallowed that fly,
Perhaps she'll die.

There was an old lady who swallowed a cow.
I don't know how she swallowed a cow!
She swallowed the cow to catch the goat ...
She swallowed the goat to catch the 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
That wiggled and wiggled and tickled inside her.
She swallowed the spider to catch the fly.
I dunno why she swallowed that fly,
Perhaps she'll die.

There was an old lady who swallowed a horse -
She's dead, of course.
```

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

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

appears several times in the output. You are to have only one **`println`** statement in your program for producing this line. The more complex redundancy has to do with pairs of lines like these:

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

and like these:

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

It is not possible to avoid this redundancy using just methods and simple **`println`** statements, so you are not expected to do so. There is, however, a structural redundancy that you can eliminate with static methods and this will be worth points. 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 should also be using static methods to capture the structure of the song. You should, for example, have a **different method** for each of **the verses** of the song (verses are separated by blank lines in the output).

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 text.

Turn in:
You should name your file **`SwallowedAFlySong.java`** and turn it in electronically on Canvas.
Before you turn in your assignment, look over the checklist at the end of this document to be sure that you have done all the right things for this assignment.

Grading Criteria:

Programming Assignments are graded as follows:

- Work/Effort: 20% for proper submission of something on time
- Correctness: 50% broken down as
  - 10 % for some resemblance to a correct solution
  - 20 % for demonstrating true understanding the problem
  - 20 % for accurate results as required by the problem and directions
- Design/Style: 30 % for well-organized work – clear design and structure, nice formatting, comments that enhance readability, etc. These factors will be explained in class and enhanced as we go through the course.

Additional explanation for the first assignment:
Program does not compile 20 points at most
Correctness means program produces the correct output.
Details include verses in the correct order, blank lines as needed, and spelling is correct.
Design/Style
Main calls one method for each verse of the song 10 points
Lines of code are indented correctly 10 points
File name is right 10 points
Program header comment 10 points

Checklist:

- ☐ I named my program **`SwallowedAFlySong`**. (I checked the spelling and capitalization.)
- ☐ This is the code I submitted to Canvas.
- ☐ There is no **`package`** statement in my code.
- ☐ There is a header comment at the top of my program:

```
// Your Name Here
// 1/13/16
// CPW 142
// Assignment #1
//
// This program will... fill in this comment yourself
```

- ☐ **`main`** does not print any lyrics, it only calls other methods.
- ☐ There is a different method for each verse of the song.
- ☐ Every verse of the song is printed by my program.
- ☐ Each method has a name that suggests what it does.
- ☐ The name of each method begins with a lowercase letter and uses CamelCase.
- ☐ Lines of code are indented correctly, 4 spaces per tab.
- ☐ I consistently used only one style of curly braces. Here are the only two styles that are accepted.

```
public static void myMethod() // vertical lineup for { and }
{
    System.out.println( "myMethod is here." );
}

public static void myMethod() { // compact lineup for { and }
    System.out.println( "myMethod is here." );
}
```