

CSE 142, Spring 2007
Programming Assignment #1: The House that Jack Built (10 points)
Due: Tuesday, April 3, 2007, 4:00 PM

Program Description:

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

```
This is the house that Jack built.

This is the malt
That lay in the house that Jack built.

This is the rat,
That ate the malt
That lay in the house that Jack built.

This is the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.

This is the dog,
That worried the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.

This is the cow with the crumpled horn,
That tossed the dog,
That worried the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.

This is the maiden all forlorn
That milked the cow with the crumpled horn,
That tossed the dog,
That worried the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.
```

The output is the first seven verses of the nursery rhyme “This is the house that Jack built” by Mother Goose. For brevity, we reduced the number of verses from the original eleven to seven. The original rhyme can be found at: <http://www.amherst.edu/~rjyanco94/literature/mothergoose/rhymes/thisisthehousethatjackbuilt.html>

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 eleven verses to match the complete nursery rhyme. You may include blank lines 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 rhyme 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 rhyme and improving the code using static methods.

Stylistic Guidelines:

Any `println` statement that prints text should not be in your `main` method. Instead, use static methods in this program, for two reasons:

1. To capture the *structure* of the rhyme's seven verses.

You should be using static methods to capture the structure of the rhyme. You should, for example, have a method for each of the seven verses of the rhyme to print that verse's entire contents. You can write additional methods as you see fit.

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

You should use only one `println` statement for each distinct non-blank line of the rhyme. 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 rhyme:

```
That lay in the house that Jack built.
```

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 rhyme and create static methods that capture those groups and are called multiple times. There is a general structural redundancy to the rhyme 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 thirteen static methods other than `main` and occupies 87 lines including comments and blank lines.

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 `print` statements (as opposed to `println`).

Submission and Grading:

Turn in your `ThisHouse.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 rhyme as specified previously.