University of Washington
Computer Science & Engineering 142 (Computer Programming I), Summer 2005

Programming Assignment #2 (Rocket)
Due: Tuesday, 7/5/2005, 6:00 PM

Program Description:

This program tests your understanding of expressions and variables, `for` loops, and global constants, and also reinforces your previous material on static methods and println statements. You should write a Java class called `Rocket` that should be saved into a file called `Rocket.java`. Your Rocket program should produce the following rocket ship figure as output:

```
     /**\
    //**\\
   ///**\\\
  ////**\\\\
 /////**\\\\\
+=*=*=*=*=*=*+
|../\..../\..|
|./\/\../\/\.|
|/\/\/\/\/\/\|
|\/\/\/\/\/\/|
|.\/\/..\/\/.|
|..\/....\/..|
+=*=*=*=*=*=*+
|\/\/\/\/\/\/|
|.\/\/..\/\/.|
|..\/....\/..|
|../\..../\..|
|./\/\../\/\.|
|/\/\/\/\/\/\|
+=*=*=*=*=*=*+
     /**\
    //**\\
   ///**\\\
  ////**\\\\
 /////**\\\\\
```

You should exactly reproduce this output.

One way to write a Java program to draw this figure would be to write a `System.out.println` statement that prints each line of the rocket. However, this solution would not receive full credit. A major part of this assignment is showing that you understand `for` loops; therefore, in lines that have repeated patterns of characters that vary in number from line to line, represent the lines and character patterns using appropriate nested `for` loops. (The complex figure in Section 2.5.8 is a good example of this.)

It may help you to write pseudo-code and tables to understand the patterns in the input.

Implementation Guidelines:

This program is intended to test your knowledge of Chapters 1 and 2, especially nested `for` loops. If you are interested, you may use the Java language features from Chapter 3, although you are not required to do so and you will receive no extra credit for doing so. You may not use any programming constructs that are not in chapters 1 through 3 of the textbook.

Continue to use static methods to structure your solution. Structure your program in such a way that the methods match the structure of the output itself. Avoid significant redundancy; make sure that no substantial groups of identical printed lines or other statements appear in your code.

You are required to properly indent your code and will lose points if you make significant indentation mistakes. See section 2.5.3 of the book for an explanation and examples of proper indentation.

Give meaningful names to methods and variables in your code. Follow Java's naming standards about the format of ClassNames, methodAndVariableNames, and CONSTANT_NAMES. Localize variables whenever possible -- that is, declare them in the smallest scope in which they are needed.

Include a comment at the beginning of your program with basic information and a description of the program and include a comment at the start of each method.

Extra Credit: Using a Global Constant for Rocket Ship Size

On this assignment, you may optionally earn 2 points of extra credit by using a global constant to represent the rocket ship's height. Your score for this program cannot exceed a 100% mark of 15 / 15 points, so an otherwise perfect program's score would not benefit from implementing this constant. But implementing the constant may give you some safety points to compensate for other potential deductions.

The various subfigures in the middle of this output have a height of 3, so 3 should be the value of your constant. These subfigures have the property that their height determines their width; therefore, you do not need a second constant to represent the width.

On any given execution your program will produce just one version of this figure. However, you should refer to the global constant throughout your code, so that by simply changing your constant's value and recompiling, your program would produce a rocket ship figure of a different size.

The course web site will contain expected output files that show you the expected output if your constant height is changed from 3 to 7. See Section 2.5.8 for an example of a program that uses a constant while drawing a complex figure.

Submission and Grading:

Name your file `Rocket.java` and turn it in electronically from the "Assignments" link on the course web page. This assignment will be worth 15 points instead of the usual 20 points.

Part of your program's score will come from its "external correctness." External correctness measures whether the output matches exactly what is expected.

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 using for loops to capture repetition in the output, capturing the structure of the figure using static methods, commenting, naming identifiers, and indentation of your source code.