

**CSCI 161: INTRODUCTION TO PROGRAMMING I**  
**LAB 4: ROCKET (20 POINTS)**  
**(VARIABLES, METHODS, for LOOPS, CONTROLLING COMPLEXITY)**

---

## Overview

**Before you do anything, read through this lab very carefully. Read the Helpful Hints.**

This program encapsulates many of the concepts we have learnt this semester. It is complex – and you need to “control the complexity”. This program should take several hours, so ensure that you give it adequate time over the week. Your textbook will be very helpful – so refer to it. Name the project “Lab4Rocket”, name the class “Rocket”.

### Instructions:

- Write a complete program that produces the figure below as output.
- Use at least 5 static methods to capture the structure and eliminate redundancy
- The methods should include for loops to eliminate redundancy within lines
- There should be no println statements in your main method, only method calls
- As always, include a comment block at the very beginning of your program with your name, date, course number, and a description of the program.
- Use good programming style: write a **comment** for each method and loop, and indent appropriately, and use white space to make your program readable. **Points will now be deducted for not following these guidelines.**
- Read the helpful hints – because they are meant to be useful

### Input Specification

No input is required

### Required Methods

Use several (ideally five) static methods, including main

### Helpful Hints

- Build a pseudocode
- Find patterns – can you see them? Divide the output into patterns. Look carefully because these will serve as your methods
- To manage the complexity, develop the program incrementally
- Express the overall structure in "main", which call the component methods
- Then proceed to the components
- For example, first write a method to draw the cone e.g. drawCone( ). Then test that method.
- Next, write a method that draws the body, noting the symmetry e.g. drawHalf1( )
- At first you may leave this method empty or draw a simple line.
- Continue your development and testing in an iterative fashion, writing small segments of code and then testing.
- This will result in significant time savings since you'll have to deal with fewer errors per test.
- Note the symmetry and the significance of the number 4 with respect to the number of lines in each pattern
- Be careful when printing backslashes
- Create tables to determine the number of dots, spaces, etc. to print for each pattern section

