1. Goal
The goal of this assignment is to better grok object-oriented programming. Start by pondering how the following text of the assignment suggests the existence of the classes polynomial and term (or perhaps nomial).

2. Problem Statement [client’s statement of their need]
Our lads, Sponge Bob and Patrick, have a polynomial system to beat the stock market as long as they can get the maths right! That means adding and multiplying polynomials.

3. Analysis  [What is the client’s problem?]  
Q: (Hey client, )What is a polynomial?  
A: Polynomial comes from poly- (meaning “many”) and -nomial (meaning ”term”) ... so it’s “many terms.” In this assignment a polynomial is the sum of powers of a variable multiplied by constant coefficients. For example, \(a_nx^n + \cdots + a_2x^2 + a_1x + a_0\).

Q: (Hey client, )What do you need to do with them?  
A: Three things: add two polynomials, multiple two polynomials, and print out a polynomial.

Q: (Hey client, )What does it mean to add two polynomials?  
A: [While I assume you know the answer, if not check out http://www.purplemath.com/modules/polyadd.htm.]

Q: (Hey client, )What does it mean to two multiply polynomials?  
A: [While I assume you know the answer, if not check out http://www.purplemath.com/modules/polymult.htm.]

Q: (Hey client, )What kind of interface do you want?  
A: None. [For this assignment, Squidward has provided a hard coded test driver!]

4. Design  [How will you as a software engineer solve this problem.]  
This is all you!

5. New Assignment Requirements [ that were not part of Assignment 3 ]  
• Create your code to be part of a package named asn5.
• You must use at least two classes from the JCF to store the terms of a polynomial. First write code using ArrayList. After you are satisfied with its performance, replace all occurrences of ArrayList with LinkedList. As necessary continue to work on the code until it works regardless of your using ArrayList or LinkedList.
• You must use generic types for the ArrayList and the LinkedList. You may also need to use generics to make terms Comparable (i.e., implement Comparable... ) and when iterating over a collection of terms.

6. Assignment Requirements [ that are the same as those of Assignment 3 ]  
• To traverse the terms of a polynomial you must use iterators. Better solution will use both the for loop version and the hasNext version.
• Use clear documentation and careful formatting. Be consistent in indentation and alignment of braces. Each open brace “{” must be on its own line.
• Each source code file must start with
  // This is my code
  // <Your Name>
  // CS312
• Finally, let’s do some empirical data analysis! As part of your test driver work out values for $K_i$ such that the ArrayList version take about 5 seconds per loop to run the loops
  ```java
  for(int i=0; i<K_1; i++)
      poly = poly + poly;
  for(int i=0; i<K_2; i++)
      poly = poly * term;
  for(int i=0; i<K_3; i++)
      poly = poly * poly;
  ```
Record in README.md the $K_i$s along with the actual time taken and the time that the LinkedList version takes for the same values of $K_i$. In a sentence or two, try to explain any differences between the times for ArrayList and LinkedList.

7. What to hand in
   (1) A well-formatted 2-up printout of your source code.
   (2) A GitHub repo that includes (you must use these names as the grading script assumes them!)
      • README.md with the sections plateau schedule, test plan, and data analysis;
      • mycode.pdf (ensure you are happy with the formatting by using xpdf); and
      • your Java source code with main in Tester.java.

8. Notes
• Consider using the testing tool JUnit to help with test automation.
• Also, consider using test-driven development: write a test, write code to pass it, repeat.
• Error checking is a plus.
• Reread the general notes regarding style, braces, and header comments.
• Wow good on you for reading this far ... one last interesting ponderable ⋯ are you better off starting from your Assignment 3 code or from scratch?