This homework will explore the issues we discussed in class related to good class design. Your grade will depend on how well you adhere to the principles of cohesion, coupling, and responsibility-based design we have discussed in class. The homework is lengthy, so get started early!

Problem 1 In this problem we will continue refining the AminoAcidSequence and the AminoAcidSequence-Set classes from Homeworks 2 and 3.

1. (20 points) Write a method called isValid() that takes a String as argument and returns true if and only if the string is a “valid” amino acid. For the purpose of this problem, a string is valid if each of its characters is a symbol for an amino acid. Which class should contain this method? Compare your solution with any solutions in previous homeworks.

2. (30 points) Implement a method called compare() that computes how many characters two AminoAcid-Sequences differ in. For instance, the result for “AAA” and “AAA” is 0 but the result for “AMMA” and “MAMA” is 2. Which class should contain this method? Will the method take one or two arguments? The method should be gracefully able to handle the case of the two sequences having different lengths.

3. (50 points) In this problem, we will simulate the process of running the popular Blast programme for sequence similarity. Implement a method in the AminoAcidSequenceSet class called runBlast() that takes an AminoAcidSequence as argument and returns the amino acid sequence stored in the AminoAcidSequenceSet that is most similar to the argument. Remember to avoid code duplication and reuse other methods you may have implemented or you can implement. Your method should return an appropriate object if there is more than one most similar sequence. Implement an overloaded runBlast method that takes an integer maxDifference as the second argument. This method will only return amino acid sequences that have at most maxDifference characters different from the argument. The two overloaded runBlast methods can share a lot of code. Make sure that you avoid writing this code twice.

Problem 2 Starting with the zuul-better project we discussed in class,

1. (30 points) solve exercises 7.20 and 7.22 from Barnes and Kölling. You will need to implement a class called Item as discussed in the book in Chapter 7.11.2. You can add items to a room in the createRoom() method in the Game class.

2. (70 points) solve exercises 7.23 and 7.26 from Barnes and Kölling. Exercise 7.26 can be difficult. You will need to find out about the Stack data structure in Java.

To solve the questions in this problem, you may find it helpful to do some of the earlier exercises in Chapter 7 of the book.

* Make sure you test all your solutions thoroughly. Consider all types of inputs a user may provide and how you will deal with them. Read Chapter 6 to refresh your memory about the types of tests you should perform.
• Do not forget to include full documentation for your source code as described in Chapter 5.10 of Barnes and Kölling.

• Create a folder called <YourName>-Homework4 and use a separate folder inside this folder for the source code for each problem.

• Submit your homework by emailing the files containing the source code to me. I prefer it if you can zip the entire folder and send the zipped folder to me. If doing so is difficult, just email me the individual files as attachments.