

Homework Reflections

A major part of developing code in the real world is getting feedback from peers and improving your code; this is known as a code review. Part of your assignment grade will be completing reflections on the feedback from your TA you receive for each assignment.

For each assignment, you will receive feedback from your TA a day or two before the next assignment is due. On that feedback, your TA will mark areas for you to improve the quality of your code, leaving annotations on how your code could be improved. We have many resources listed that detail style issues that are common when first learning programming in Java (Style Guide, Commenting Guide, General Style Deductions, the homework specifications themselves, etc.) and how to avoid them.

Your job is to address at least 3 annotations from your previous assignment. You should copy-and-paste the feedback your TA gave you, find a relevant resource that detailed why that was a mistake, and copy-and-paste the snippet from the resource that detailed it's mistake. If you did not make at least 3 mistakes, you should post up questions that you ran into as you were completing the current assignment until you hit 3 points.

This reflection will get you to reflect on your past errors, get acclimated to the resources available for you to succeed, and create the opportunity for more feedback and understanding of style. Once you have completed this reflection, make sure that you look through the assignment you will be turning in and update your code to avoid those same mistakes!

These files should be turned in a file called `reflection.txt`. Note that for .txt files, you need to put in your line-breaks manually so everything's not on one single line.

Below we have an example template of how to structure your reflections.

Reflection Template

1.TA Feedback:

“Note that the style guide states that we should minimize the number of fields whenever possible to keep the classes we create simple”

Resource:

Style Guide

Snippet from Resource:

“The more fields your class has, the more difficult it becomes to maintain and reason about your code. With more fields, each instance of the class will also take up more memory.”

2.TA Feedback:

“Note that talking about internal fields such as “elementData” is an implementation detail that clients don’t need to know about”

Resource:

Commenting Guide

Snippet from Resource:

“They will not see the insides of your methods, nor will they see anything private like private fields or private helper methods. Talking about those things is considered implementation details.”

3.TA Feedback:

“It’s great that you documented that an exception would be thrown if the passed in integer was less than 0, but we need to also document the exact type of exception that is thrown”

Resource:

General Style Deductions

Snippet from Resource:

“method header does not describe exceptions that are thrown when preconditions are violated, including the specific type of exception and the conditions under which it is thrown”