# Homework 0

## Arik Rinberg 317156669

## Barak Zan 00000000

# Question 1 – Coding Style Guidelines Naming Conventions

### Package naming conventions

Package names should be all lowercase, with consecutive words with no separation. For example and **not** or .

### Class naming conventions

Class names should be upper camel case, without underscores. The class should be descriptive without shortening the words. For instance a class name for asynchronous socket handling should be and not .

### Function name conventions

Method names should be lower camel case for non-static methods, and upper camel case for static methods, and should be descriptive. For instance "" and not "".

### Variable name conventions

Variables should be lower camel case and will be nouns. The variable names should be descriptive of what they are, and not abbreviated. For example, "" and not "" or "".

Private variables will start with an underscore.

Variables representing constants should be all uppercase such as "".

## Line styling and indentations

### Line length

The length of a line will not exceed 120 characters.

### Indentation and curly brackets

Line indentation will be done with tabs to represent the scopes. Open curly brackets will be opened in a new line.

For instance:



Blocks in the code will separated by a new line.

## Comments and Documentation

Above every class and method should be a clear documentation.  
Documentation of methods must include specifications as described in the course.

Comments should be added to the code when the code isn't self-explanatory.

# Question 2 – Comment Reader

## Input to CommentReader

## Output of CommentReader



# Question 3 – Ball Container

## Different implementations for getVolume()

Our first implementation to getVolume() was:

Our second implementation to getVolume() was holding a variable in the class and updating it on every add and remove, and zeroing the variable when clearing the container.

The first implementation is better if the calls to getVolume() are infrequent. As there is no extra work to be done on update and remove, on the rare occasions we call getVolume(), iterating over the ArrayList won't matter so much. Also this implementation uses less memory than holding an extra variable.

However, if the calls to getVolume() are frequent, we don't want to waste time constantly iterating over the ArrayList, so using an updated variable is more efficient.

## Change to @requires for add()

### Needed alterations to the specification

The addition of the @requires doesn't need changes to the specification, as nowhere in the specification referenced what happens if the parameter ball was null.

### Needed alterations

The addition of the @requires doesn't need changes to the implementation. We can now forgo the check if ball is null but we don't have to.

### Strength of specification after alteration

The specification is now weaker, as the @requires is longer and the @effects remains the same.

# Question 4 – Multiple choice

## Choice between two specifications

The correct answer is א, it cannot be established which specification is stronger. For a specification to be stronger than a specification we require that (1) the @requires be stronger and (2) the @effects be more specific. In the question has a stronger @requires but has a stronger @effects, so we cannot establish a stronger specification.

## Specification of the method find()

### implements

The correct answer is ד, in this case also implements and . It implements because they have the same effect, and it implements because if the color is the list it will return the location of the color. It does not implement on account of the error thrown if the color is not found.

### implements

The correct answer is ו, the implementation only implements . The implementation may rely on the fact that the list is ordered in order to find the location. For the list may be not ordered, and therefore the implementation may return the wrong value. For , the specification throws an error if not found, therefore doesn't implement it. Finally for , in the same way as the function may return the wrong value as the list may be not ordered.

### implements

The correct answer is ג, the implementation also implements . For , if color is in the list the location will be returned which is the only return value specified. For and , if the color isn't in the list, the implementation will throw an error rather than return -1, so doesn't implement these two specifications.

### implements

The correct answer is ו, the implementation only implements . The specification doesn’t specify what happens if color isn't in the list, therefore the implementation may return the value which is not as specified in all the other specifications.