

## Homework 10

### Q1)(E)

The code compiles but throws an exception at runtime. `sb` variable is initialized with an empty `StringBuilder` object. Then the program arguments are iterated. For every given argument, the loop tries to insert the `String` element being iterated to the `sb` `StringBuilder` at the index of character `'c'` in `sb`. But since the `sb` `StringBuilder` is empty, `sb.indexOf("c")` returns `-1`, and hence the code throws `StringIndexOutOfBoundsException` at runtime.

### Q2)(C, E)

Addition operator has higher precedence over assignment operator, so option A is wrong. For option B, addition operator is listed last, but it has lower precedence over the previous three operators, so it is wrong too. Option D is wrong because the subtraction operator has lower precedence over the multiplication operator. Option C and option E have the correct order of operators according to increasing or the same level of precedence.

### Q3)(C, B, F)

Option A is a getter method but it has a parameter in its method signature, so it is not a valid JavaBean. Option D should return a boolean, looking at the name, as it is prefixed with `'is'`, so this option is not a valid bean too. Lastly, Option E is wrong because there is no such naming convention. C, B, F are valid beans.

### Q4)(A, E)

`Array` has `length` variable instead of a `size()` method. `size()` method is used for `ArrayLists`. Upon the correction of this one line, the code would compile and run without issue. The nested for loop iterates correctly, and without trying to access out of bounds of the crossword array. It only iterates through the half of every array in the second dimension though. Because the second dimension has `10` arrays in it with every array having length `20`, we are iterating every `10` array but stopping after the ninth element which corresponds to half of the whole elements. And the unassigned half has its values set to the default value `0`.

### Q5)(B, D)

If a file system resource becomes temporarily unavailable, a checked exception must be used. `Error` is not a subclass of the class `Exception`, but `Throwable` class. If a user enters invalid input, I think an unchecked exception must be used, because we can consider it a programming error. Although it is possible, it is very unlikely that we would want to catch it.

$\mathbb{Q}()$

Q()  
Q()  
Q()  
Q()