

Tips for taking the AP test

(or how not to get had!)

Most of the Computer Science A questions are a pretty good gauge of how good you have become at programming in Java. As with any multiple-choice test, though, not knowing what to look out for can unfortunately heavily impact your score. :(

Write out variables (especially data structures) and their contents

If you come across an array method, for example, write out a grid representing an array and update its values as you read through the code. This helps a lot with index trickery, always/never questions, etc.

Beware of index trickery with arrays and ArrayLists!

If a for loop body involves adding or subtracting from the loop variable which is used as an index, there should be a corresponding adjustment in the loop update. Writing out your variables will help find these OBOBs (off by one bugs).

```
for(int i = 0; i < arr.length; i++) { // bound should be arr.length - 1
    if(arr[i] < arr[i + 1]) // BAD!! IndexOutOfBoundsException
        // more stuff
}
```

Look for words like 'always' or 'never'

Some questions will ask you which of a series of method calls will ALWAYS, for example, cause an error.

Make sure you test all cases (For example: empty array, single element array, etc). Write out your test data and run through each case fully.

Pay close attention to the order of statements

Printing then updating is not the same as first updating then printing. Write out your variables and read code sequentially. Sometimes the code you see on the test doesn't actually make much sense so you just need to go line by line and try to ignore your expectations.

Beware of nested calls

mystery(mystery(2)) is not the same as mystery(2)

Beware of missing curly braces

Recall that indentation is just for human readability in Java! In the code below, foo++ executes whether or not foo == bar because there are no curly braces around it.

```
if(foo == bar)
    bar ++;
    foo ++;
```

Beware of conditional operators

The opposite of > is <=, not <

Remember deMorgan's law

!(p || q) is the same as !p && !q

!(p && q) is the same as !p || !q

Take time to refer to the Appendix!

There's not a ton of value to you memorizing anything from Gridworld or even the exact behavior of substring -- knowing where to look things up is a much more important skill for you to learn.