COM S 227
Fall 2019
Miniassignment 1: Loops and Arrays
40 points

Due Date: Thursday, October 24, 11:59 pm (midnight)
5% bonus for submitting 1 day early (by 11:59 pm October 23)
10% penalty for submitting 1 day late (by 11:59 pm October 25)
No submissions accepted after October 25, 11:59 pm
Note: This is a miniassignment and the grading is automated. If you do not submit it correctly, you will receive at most half credit.

General Information

This assignment is to be done on your own. See the Academic Dishonesty policy in the syllabus, http://www.cs.iastate.edu/ cs227/syllabus.html#ad , for details.
You will not be able to submit your work unless you have completed the *Academic Dishonesty policy acknowledgement* on the Homework page on Canvas. Please do this right away.
If you need help, see your instructor or one of the TAs. Lots of help is also available through the Piazza discussions.

Please start the assignment as soon as possible and get your questions answered right away. It is physically impossible for the staff to provide individual help to everyone the night that the assignment is due!

Tips from the experts: How to waste a lot of time on this assignment

- Start the assignment the day it's due. That way, if you have questions, the TAs will be too busy to help you and you can blame the staff when you get a bad grade.
- Don't bother reading the rest of this document, or even the specification, especially the "Getting started" section. Documentation is for chumps.
- Don't test your code. It's such fun to remain in suspense until you get your score!
- The main guiding principle is: *Try to write all the code before you figure out what it's supposed to do.*

Overview

This is a short set of practice problems involving writing loops and using arrays. You will write eight methods for the class LoopsAndArrays. All of the methods are static, so your class will not have any instance variables (or any static variables, for that matter). There is a constructor, but it is declared private so the class cannot be instantiated.

For details and examples see the online javadoc.

My Code's Not Working!

Developing loops can be hard. If you are getting errors, a good idea is to take a simple concrete example, and trace execution of your code by hand (as illustrated in section 6.2 of the text) to see if the code is doing what you want it to do. You can also trace what's happening in the code by temporarily inserting println statements to check whether variables are getting updated in the way you expect. (Remember to remove the extra println's when you're done!)

Overall, the best way to trace through code with the debugger, as we are practicing in Lab 6. Learn to use the debugger effectively, and it will be a lifelong friend.

Always remember: one of the wonderful things about programming is that within the world of your own code, you have absolute, godlike power. If the code isn't doing what you want it to do, you can decide what you really want, and make it so. You are in complete control! (If you are not sure what you want the code to do, well, that's a different problem. Go back to the "Advice" section.)

Specification

The specification for this assignment includes this pdf, the `LoopsAndArrays` javadocs, along with any "official" clarifications posted on Piazza.

Where's the main() method??

There isn't one! Like most Java classes, this isn't a complete program and you can't "run" it by itself. It's just a single class, that is, the definition for a type of object that might be part of a larger system. To try out your class, you can write a test class with a main method.

There will also be a SpecChecker (not ready yet) that will perform a lot of functional tests, but when you are developing and debugging your code at first you'll always want to have some simple test cases of your own as in the main method above.

Note that neither the SpecChecker nor the provided test program nor the combination of the pair provide complete, comprehensive test coverage of your implementation. You will have to write your own tests.

Sample usage

A good way to think about the specification is to try to write some simple test cases and think about what behavior you expect to see.

There is also a SpecChecker (see below) that will perform a lot of functional tests, but when you are developing and debugging your code at first you'll always want to have some simple test cases of your own.

Suggestions for getting started

Smart developers don't try to write all the code and then try to find dozens of errors all at once; they work *incrementally* and test every new feature as it's written. Here is a rough guide for how an experienced coder might go about creating a class such as this one:

- Create a new, empty project and add a package called `mini1`.
- Create the `LoopsAndArrays` class in the `mini1` package and put in stubs for all the required methods. For methods that are required to return a value, just put in a "dummy" return statement that returns zero or `false`.
- Download the specchecker, import it into your project as you did in labs 1 and 2, and run it. There will be lots of error messages appearing in the console output, since you haven't actually implemented the methods yet. Always start reading from the top. All you really want to check at this point is whether you have a missing or extra public method, if the method declarations are incorrect, or if something is really wrong like the class having the incorrect name or package. Any such errors will appear first in the output and will usually say "Class does not conform to specification."
- Before you write code for a method, always write a simple usage example or test case. This will make sure you understand what the code is really supposed to do, and it will give later you a way to check

whether you did it correctly. Of course, if you are really not sure what a method is supposed to do, bring up your question for discussion on Piazza!

The SpecChecker

You can find the SpecChecker online; see the Piazza Homework post for the link. Import and run the SpecChecker just as you practiced in Labs 1 and 2. It will run a number of functional tests and then bring up a dialog offering to create a zip file to submit. Remember that error messages will appear in the console output. There are many test cases so there may be an overwhelming number of error messages. *Always start reading the errors at the top and make incremental corrections in the code to fix them*. When you are happy with your results, click "Yes" at the dialog to create the zip file. See the document "SpecChecker HOWTO", which can be found in the Piazza pinned messages

More about grading

This is a miniassignment and the grading is automated. If you do not submit it correctly, you will receive at most half credit.

If you have questions

For questions, please see the Piazza Q & A pages and click on the folder `miniassignment1`. If you don't find your question answered, then create a new post with your question. Try to state the question or topic clearly in the title of your post, and attach the tag `miniassignment1`. *But remember, do not post any source code for the classes that are to be turned in*. It is fine to post source code for general Java examples that are not being turned in. (In the Piazza editor, use the button labeled "pre" to have Java code formatted the way you typed it.)

If you have a question that absolutely cannot be asked without showing part of your source code, make the post "private" so that only the instructors and TAs can see it. Be sure you have stated a specific question; vague requests of the form "read all my code and tell me what's wrong with it" will generally be ignored.

Of course, the instructors and TAs are always available to help you. See the Office Hours section of the syllabus to find a time that is convenient for you. We do our best to answer every question carefully, short of actually writing your code for you, but it would be unfair for the staff to fully review your assignment in detail before it is turned in.

Any posts from the instructors on Piazza that are labeled "Official Clarification" are considered to be part of the spec, and you may lose points if you ignore them. Such posts will always be placed in the Announcements section of the course page in addition to the Q & A page. (We promise that no official clarifications will be posted within 24 hours of the due date.)

What to turn in

Please submit, on Canvas, the zip file that is created by the SpecChecker. The file will be named SUBMIT_THIS_mini1.zip. and it will be located in the directory you selected when you ran the SpecChecker. It should contain one directory, mini1, which in turn contains one file, LoopsAndArrays.java. Always LOOK in the zip file the file to check what you have submitted! Submit the zip file to Canvas using the Miniassignment1 submission link and verify that your submission was successful. If you are not sure how to do this, see the document "Assignment Submission HOWTO" which can be found in the Piazza pinned messages under Syllabus, office hours, useful links.

We strongly recommend that you just submit the zip file created by the specchecker. If you mess something up and we have to run your code manually, you will receive at most half the points. If necessary for some reason, you can create a zip file yourself. The zip file must contain the directory mini1, which in turn should contain the file LoopsAndArrays.java. You can accomplish this by zipping up the src directory of your project. The file must be a zip file, so be sure you are using the Windows or Mac zip utility, and not a third-party installation of WinRAR, 7-zip, or Winzip.