Project #4
CS 2210 – Fall 2021
Sam DeCook

I. Requirements: Restate the problem specification, and any detailed requirements

Develop an ArrayQueue class which implements the Queue interface. Test with JUnit.

II. Design: How did you attack the problem? What choices did you make in your design, and why? Show class diagrams for more complex designs.

The main work was done when considering how to make an array function like a queue, thankfully, that was done in class.

III. Security Analysis: State the potential security vulnerabilities of your design. How could these vulnerabilities be exploited by an adversary? What would be the impact if the vulnerability is exploited?

There really aren't any vulnerabilities, this is a pretty simple class.

IV. Implementation: Outline any interesting implementation details.

To wrap the array so we didn't waste space, we used the mod operator.

V. Testing: Explain how you tested your program, enumerating the tests if possible. Explain why your test set was sufficient to believe that the software is working properly, i.e., what were the range of possibilities of errors that you were testing for.

I tested all the methods in a basic use case, expanding the array when it was wrapped. Then I tested expanding when it was wrapped and finally tested the exceptions.

VI. Summary/Conclusion: Present your results. Did it work properly? Are there any limitations? If it is an analysis-type project, this section may be significantly longer than for a simple implementation-type project.

The array queue worked properly.

VII. Lessons Learned: List any lessons learned. What might you have done differently if you were going to attack this again.

Just where and where not to put the <E> parameter.

My code compiled and ran properly, and produced the expected output.