CSSSKL 143 Lab: Debugging and Exceptions

**Using a Debugger**

Answer the following questions in your comments in any file and submit that.

1. Using your debugger, debug Account.java and DebuggingExercise4.java
2. Hover over variables to see their values.

>>> I am using intellij and it provides a guided step by step approach in debugging where showing the dependencies of the classes as well as allowing to step over.

1. Find the shortcut keys for the following debugging commands, and describe what each does in comments.

The closest is the ALT+F9 that jumps to the cursor.

F8 Step over advances the execution one step forward, but it doesn't visit any method

F7 Step into - step into a method

* 1. Continue – continues the execution flow
     1. What does this do?
     2. How is it different than moving in steps?

**Debugging With Asserts**

1. Look at and explain the first two asserts in main; which of the two is breaking the program?
   1. Comment out the offending assert line and rerun your code

>>>Assert(false) is failing since false is true and not false;

* 1. Find the checkAddress() function and explain in comments what it does
     1. To what address does “this” map to?

>>>AssertDemo@4439f31e

* + 1. To what address does “input” or “ad“ map to?

>>>AssertDemo@4439f31e

* + - 1. What does this mean? Explain this in your comments.

>>>This is an object that has been created in the AssertWithObjects and passed to checkAddress, basically it’s the same as input

* 1. At the end of this subroutine, build two new asserts that test Objects.
     1. How would we test immutable objects? How is this similar to primitives?

>>>Immutable objects are tested via equals function but primitives can be tested via ==;

* + 1. When building these new asserts, consider testing Objects from previous labs, such as a Point2D or a Date.

>>> I have created a assert (a.equals(b));

**Reflection and Analysis Questions**

Put the answers to this questions in comments inside of your code (AssertDemo.java). Of the debugging techniques covered here...

1. Which technique do you prefer? Why?

>>>I like using intellij debugger as it’s providing a good flow of events that need to be fixed

1. Are some debugging techniques more (or less) appropriate for longer programs?

>>>More or less 😊 since focusing on just one method > class > group would help to ensure those are executed correctly. If there are dependencies, stepping over the methods may also help.

1. What are the advantages to using a debugger with a GUI?
   1. What can you inspect here that you couldn’t when just printing to the console?

>>>Maybe visual results?

1. How can proper documentation help in finding and avoiding bugs?

>>> By giving a heads up on the system design and proper implementation