**Overview of Report Assignment:**

* Software for training accountants and analysts
* Assess quality of the software of its freedom from defects. (define what this means)
* Assume that the behaviour of the software is consistent with the manufacturer’s description of the HP-12C, such as the user manual (find and read this)
* If anything is ambiguous, I should justify my specification assumptions in terms of either general mathematics or financial calculations conventions.
* Read pages 31-35 in the Kaner et al book for advice on deriving requirements as part of the testing process.

**SECTION A – An Overview**

Define what constitutes ‘the software under test’; listing the features that you will test and you will not test.

Explain how you have determined the expected behaviour of the software (in the absense of an exhaustive and explicit requirements specification).

Explain the prioritisation strategy – which user needs and concerns, within the broader requirements, you have treated as most important

Explain the overall strategy you have used for creating test cases, and for selecting specific test cases that you present in section B

Define acceptability criteria for the software – what(testable) properties does it need to have in order for it to be of acceptable quality for its intended purpose?

* Correct results;
* Correct graphical user interface display

**SECTION B – Test Cases**

**Notes:**

* **Each test case should only test one thing – one feature, unusual input or one user task.**
* **Diverse range of test cases to cover different aspects.**
* **Choose the most interesting ones.**
* **Test at either unit, integration and system level. System level is sufficient.**
* **Short fragments of code are only allowed in the report.**
* **If something is true for all test cases, say this once at the start of the section.**

***@Test 1:***

* *State the purpose of the test within the test set. This is vital. Instead, state the question that the test case asks the software.*
* *Describe the stimulus applied to the software (which API calls, and the expected results).*
  + *Define contraints*
  + *The reason why the result is the way it is*
* *Document results that occurred when the test cases are run. You should provide explicit indication of whether each test passed or failed, and in the latter case state what happened instead.*

@Test 1: [name]

* In this case,
* [Testing description]As apposed to scripted testing which is directed from requirements, this more exploratory test aims to be more investigative, where the tester’s mind is in control. The determination of the test cases was made during testing, as opposed tobeing made in advance.
* I aimed to see this first-hand from the end-user’s perspective, and not as a developer. Therefore, I used a 3rd party candidate to perform this test. Some notes on the candidate:
  + This was the perfect candidate she has a lack of understanding for mathematics and finance, but is still interseted enough to interact with the program.
  + Whilst my niece was able to identify and make basic interactions with the program, she was more limited by…
  + Has an accute understanding of basic computer functionality such as key strokes and mouse-clicks.
  + Is able to identify and make basic interactions with the
  + Is two years of age.
* The stimulus for this was… Constraints involved:
  + Involved no reference to the HP 12C User Manual or the README.pdf from the downloaded software package
  + Ran the .jar file straight from the package, as opposed to importing it into an IDE like IntelliJ or Eclipse, giving no insight into the program’s development (could not see the code, the package structure, the engineering – the inside working of the program – **what is the technical word for this?)** Essentially I
  + Even though the test involved keys on the calculator being pressed, the goal was not to perform mathematical or financial calculations. It was purely to experiment with its functionality. This could include
  + To better the results, I set up the application to the end-user
  + Limited to 10 operations – more specifically, 10 mouse-clicks. Every step noted.

Result:

The most interesting test case performed revealed an error in which…

@

**SECTION C – Summary**

* **A summary of the testing performed**
* **A summary of the results observed, including classification of the faults found**
* **An overall evaluation of the thoroughness and quality of the tsting you have performed**
  + **What might be possible given substantial resources there were**
* **An overall evaluation of the software tested in terms of its freedom from faults**