**Christine Hainly**

Simple Java Login With JUnit Testing

**06/31/2018**

# Table of Contents:

Contents

[Table of Contents: 2](#_Toc518146984)

[Part 1: Creating a Login Graphical User Interface – 3](#_Toc518146985)

[Figure 1.1 – Login Screen 3](#_Toc518146986)

[Figure 1.2 – Correct Login Pop-up 3](#_Toc518146987)

[Figure 1.3 – Incorrect Login Pop-up 3](#_Toc518146988)

[Part 2: JUnit Testing of the Login and Log Functions – 4](#_Toc518146989)

[Figure 2.1 – Setup() and Teardown() Functions 4](#_Toc518146990)

[Figure 2.2 – testWriteNewEntry() Function 4](#_Toc518146991)

[Figure 2.3 – testMain() Function 5](#_Toc518146992)

[Figure 2.4 – testInitialize() Function 5](#_Toc518146993)

[Figure 2.5 – testAuthenticate() Function 5](#_Toc518146994)

[Figure 2.6 – Test Results 6](#_Toc518146995)

[Conclusion: 6](#_Toc518146996)

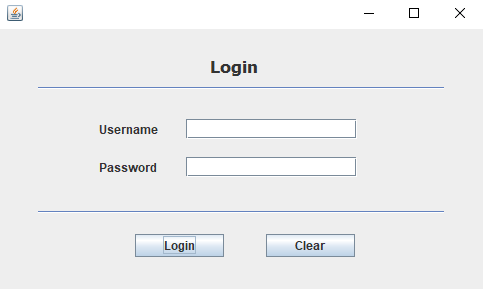
[Figure C.1 – OK Prompt 7](#_Toc518146997)

[References: 8](#_Toc518146998)

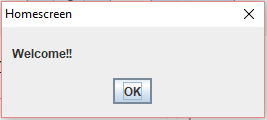
# Part 1: Creating a Login Graphical User Interface –

I created a Graphical User Interface (GUI) for this project, in NetBeans, that allows the user to enter in a username and password. The GUI had the option to enter or clear the text input on the window. Figure 1.1 shows the login screen and buttons as explained above. If a valid username and password were entered a box would pop up saying “Welcome!” as shown in Figure 1.2. If invalid user information was entered a box would pop up telling the user that “Incorrect Login Information” was entered, as shown in Figure 1.3.

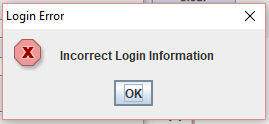
## Figure 1.1 – Login Screen



## Figure 1.2 – Correct Login Pop-up



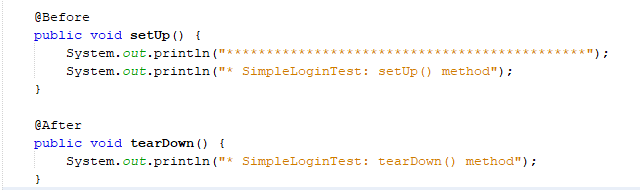
## Figure 1.3 – Incorrect Login Pop-up



# Part 2: JUnit Testing of the Login and Log Functions –

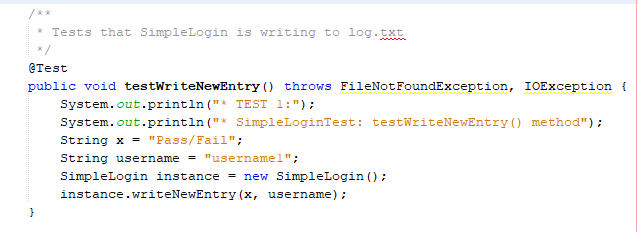
I followed and completed the tutorial “Writing Junit Tests in NetBeans IDE. After doing the tutorial I started creating a unit test for my own simple login project. Figure 1 through Figure 5 shows each section of the SimpleLoginTest.java file I created.

## Figure 2.1 – Setup() and Teardown() Functions



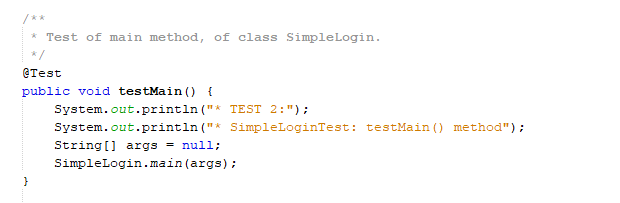
## Figure 2.2 – testWriteNewEntry() Function

This function sets the variables “x” and “username” and enters them into the writeNewEntry() function. This prints out a test log entry which passes.



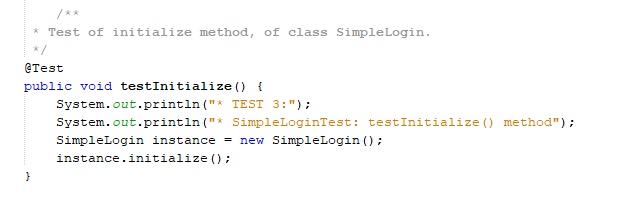
## Figure 2.3 – testMain() Function

This function tests the main() functions arguments.



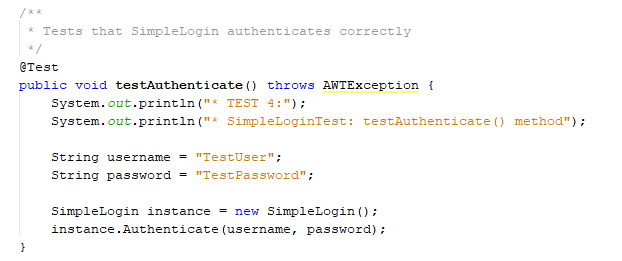
## Figure 2.4 – testInitialize() Function

This function tests the initialize() function to make sure it runs properly.



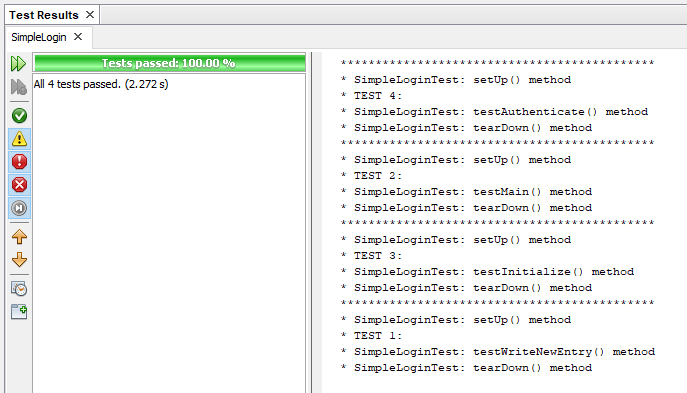
## Figure 2.5 – testAuthenticate() Function

This function tests the authenticate() function. I have set the username and password variables and enter them manually into this instance of the authenticate() function which passes.



Once all the tests were written correctly I was able to run them all together and verified that they all passed. Figure 2.6 shows the results of the test file I created using JUnit.

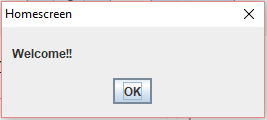
## Figure 2.6 – Test Results



# Conclusion:

I thought that writing unit tests would be extremely easy, but learned quickly that when a GUI is involved the testing becomes more complex. I also had trouble writing a test for login authentication with the original why I had programmed the GUI. This caused me to rewrite the code to incorporate an Authenticate() function that was able to be included in the test. I also encountered a problem when I ran my final test file and was not yet able to come up with a solution to. When I run my test it runs fine until it reaches testAuthenticate(). The correct login information is entered which causes a popup window prompting the user to click OK, as shown in Figure C.1.

## Figure C.1 – OK Prompt



Once I hit OK the test would finish and pass. I searched for a way around this and found a resource stating that the current version of JUnit doesn’t work well with GUI’s. I am not sure how accurate that is, but I am working on finding a way to interact with a GUI using JUnit.

# References:

Dimtsa, K. (2018, June 16). JUnit Tutorial for Unit Testing – The ULTIMATE Guide (PDF Download). Retrieved from <https://www.javacodegeeks.com/2014/11/junit-tutorial-unit-testing.html#unit_testing_java>

Ganfield, K. (n.d.). Writing JUnit Tests in NetBeans IDE. Retrieved from <https://netbeans.org/kb/docs/java/junit-intro.html>

(2014, March 12). Retrieved June 30, 2018, from <https://www.youtube.com/watch?v=1M7gzXC9434>