**Homework Assignment 1 (100 Points)**

**CSE 464/598, Fall 2015**

**CIDSE, Arizona State University**

**Due: By Sep 19th 11:59 pm**

**Introduction:** This assignment helps you to reinforce the topics discussed in the class including

1. Basic terminology of Testing and Quality Assurance, Software Quality, Software Quality Assurance, and Intro to Testing.
2. Categories of testing and equivalent class testing.
3. Writing Junit test cases for test automation
4. **[50 Points]** One of functionalities of a password generator/verifier is to determine if a password user selected is strong enough.

a) Two essential password rules:

Following two rules are bare minimal that you should follow while creating a password.  
**Rule 1 – Password Length:** Stick with passwords that are at least 8 characters in length. The more character in the passwords is better, as the time taken to crack the password by an attacker will be longer. 10 characters or longer are better.

**Rule 2 – Password Complexity:** Should contain at least one character from each of the following group. At least 4 characters in your passwords should be each one of the following.

1. Lower case alphabets
2. Upper case alphabets
3. Numbers
4. Special Characters

We can call the above two rules combined as **“8 4 Rule”** (Eight Four Rule):

* **8** = 8 characters minimum length
* **4** = 1 lower case + 1 upper case + 1 number + 1 special character.

(Source: http://www.thegeekstuff.com/2008/06/the-ultimate-guide-for-creating-strong-passwords/)

Suppose you are using equivalent partitioning technique for testing if the password user selected is acceptable.

1. Identify equivalent partitions based on the above description
2. Develop test cases in a tabular format given below based on your equivalent partitions above.

|  |  |  |  |
| --- | --- | --- | --- |
| Test case # | Partition Tested | Input(s) | Expected output |

**2.[ 50 Points]** Equivalent Partitions: Finding roots of a quadratic equation

a) [25 Pts] Study the Roots.java and usingRoots.java programs given. Roots.java program finds the roots of a quadratic equation in the form ax^2 + bx + c=0. If you are to design test cases to test the Roots.java program using equivalence partitioning:

i) Identify equivalent classes to test the Roots program. Briefly describe your strategy first.

ii) Design test cases based on your equivalence classes.

**For this part you need submit**

Equivalent partitions and test cases based on your equivalent partitions. Test cases should be in a tabular format discussed in the class.

**b)[25 Pts]** Develop JUnit test cases test to your program based on test cases above (b) **for equivalent partitions**. Develop a Junit test program based on equivalent partitions. Run your test cases and verify your program is error free.

For this part you need submit

1. Junit test programs with test cases implemented.

**Submission Instructions:**

*Things to submit:*

1. Answers to question 1 part a and b typed as a text (word) document . Handwritten is okay if readable and clear.

For question 2:

1. Clearly identified equivalent partitions and test cases (answer to part a)
2. Junit test programs (answer to part b)

*How to submit:*

Create a zip file that contain 4 submission items listed above and submit online to the blackboard.