Workshop 1 (Week 2) - Black Box Testing

The purpose of this workshop is to practice and develop an understanding of Black Box testing.

1. Concepts

I. What relationship is there between equivalency classes for a given system?

II. What inputs should be chosen when performing Equivalence Partitioning?

III. What inputs should be chosen when performing Boundary Value Analysis?

2. Password Tester

Based on the documentation for the PasswordTester class (see Appendix), answer the following questions:

Considering both the return value of PasswordTester.isStrong(String)
and messages it may print to standard output:

IV. How many equivalence classes would there be for this system?

V. What are the equivalence classes?

VI. What inputs would be required to test these classes?

VII. What boundaries, if any, could be considered for Boundary Value Analysis?

3. Other Examples

Describe (in words) at least one simple system to which Equivalence Partioning and Boundary Value Analysis can be applied, and answer the following questions about that system:

I. How many equivalence classes would there be for this system?

II. What are the equivalence classes?

III. What inputs would be required to test these classes?

IV. What boundaries, if any, could be considered for Boundary Value Analysis?

4. Try the Web: Code In Game

https://www.codingame.com/ide/puzzle/power-of-thor-episode-1

Appendix: Class PasswordTester (in JavaDoc format)

public class **PasswordTester**
extends java.lang.Object

- ***Constructor Summary***

Constructors

Constructor and Description
PasswordTester()

- ***Method Summary***

All Methods **Static Methods** **Concrete Methods**

Modifier and Type	Method and Description
static boolean	isStrong(java.lang.String password) Tests the strength of a candidate password against several criteria.

- ***Constructor Detail***

- **PasswordTester**

public PasswordTester()

- ***Method Detail***

- **isStrong**

public static boolean isStrong(java.lang.String password)

Tests the strength of a candidate password against several criteria. For each criteria which has not been met, a notice will be displayed via standard output.
For the purposes of this test a strong password must have at least:
- A length of 8 or more characters
- 1 lower case letter
- 1 upper case letter
- 1 number
- 1 special character from the following list: !, @, #, $, %, ^, &, *, (,)

Parameters:

password - a password to test

Returns:

true if the password meets all of the above criteria, false otherwise.