**Montgomery College**

**CMSC 204**

**Assignment 1**

1) Write the pseudo code for the following methods:

isValidPassword, isWeakPassword, invalidPasswords

Declare static Boolean password = False

Create Public Boolean static type function isValidePassword that takes the String passwordString as a parameter

Throws LengthException, NoDigitException, NoUpperAlphaException, NoLowerAlphaException, NoSpecialSymbolException, and IncalideSequenceException

If passwordString length is less than 6

Throw LengthException and return False

If passwordString contains no digit

Throw NoDigitException and return False

If passowrdString contains no upper-case letter

Throw NoUpperAlphaException and return False

If passwordString contains no lower-case letter

Throw NoLowerAlphaException and return False

If passwordString contains no special characters

Throw NoSpecialSymbolException and return False

Loop over passwordString

If passwordString contains more than 2 continuous repeated characters

Throw InvalideSqeunceException and return False

Return True

Create Public Boolean static type function isWeakPassword that has a String passwordString parameter.

Throws WeakPasswordException

If passwordString is between 6 and 9 characters

Throw WeakPassowrdException and return True

Return False

Create Public static ArrayList<String> getInvalidePasswords function that takes passwords as ArrayList in the parameter

Create a new ArrayList<String> illegalPasswords.

Use for each loop to loop passwords

If passwords is not null

Add the password checked in the passwordCheck method to illegalPasswords

Return illegalPasswords

Create a Private String static type passwordCheck that takes String password as parameter

Declare String message = null

Try:

If the function isValidePassword True

Return empty String

Catch:

LengthException and printout message

Catch:

NoDigitException and printout message

Catch:

NoUpperAlphaException and printout message

Catch:

NoLowerAlphaException and printout message

Catch:

NoSpecialSymbolException and printout message

Catch:

InvalideSequenceException and printout message

Return message

Create classes for LengthException, NoDigitException, NoUpperAlphaException, NoLowerAlphaException, NoSpecialSymbolException, IncalideSequenceException, WeakPasswordException.

Class LengthException (the rest of exceptions above will have the same structures as this class, meaning just replace the name LengthException to other exceptions)

Declare final Boolean value

LengthException( Boolean value)

Super(Message to print)

set this.value to value

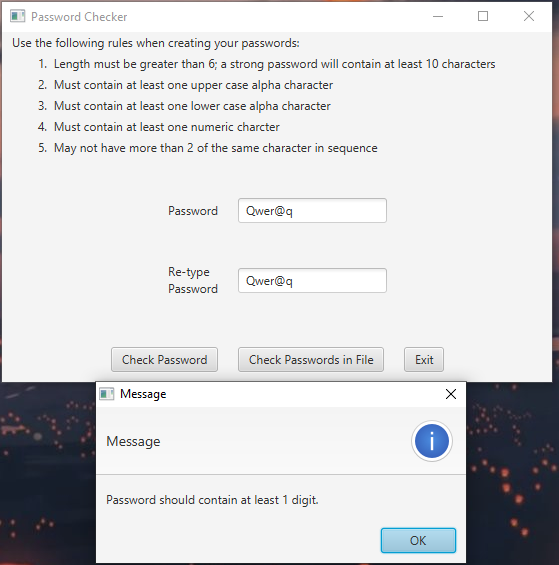
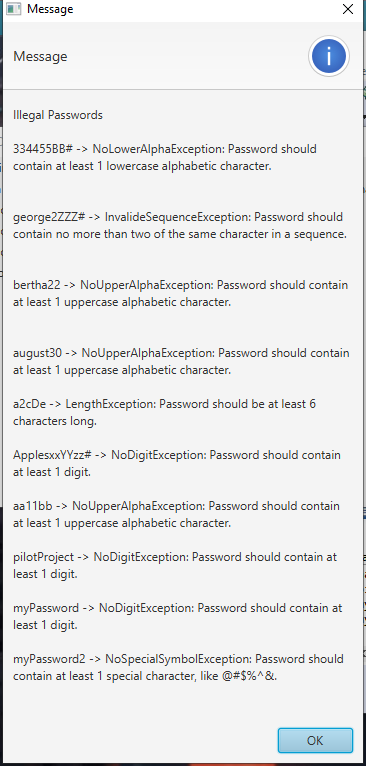
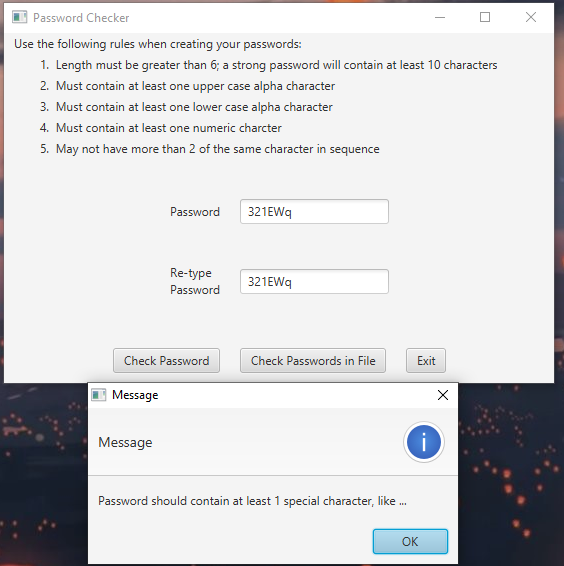
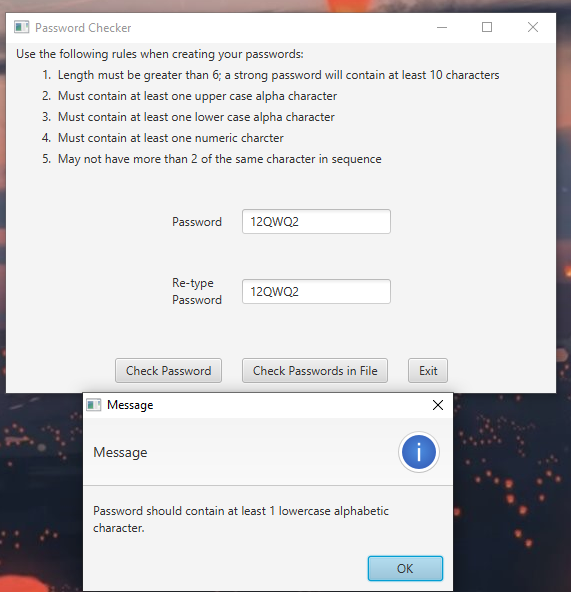
Create getValue function (Boolean value)

Return value

2)Complete the following test table. At this point you only need to complete the **Input** and **Expected** **Output** columns. Later when the implementation is complete, you will complete the **Actual Input** and **Actual Output** columns and compare them to see if the tests passed or not.

Make sure your tests cover all the possible scenarios.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Actual Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | 12QWQ2 | 12QWQ2 | Password should contain at least 1 lowercase alphabetic letter | Password should contain at least 1 lowercase alphabetic letter | Yes |
| 2 | Qwer@q | Qwer@q | Password should contain at least 1 digit | Password should contain at least 1 digit | Yes |
| 3 | 321EWq | 321EWq | Password should contain at least 1 special character | Password should contain at least 1 special character | Yes |



Reflection:

Although Assignment 1 has some changes, the main idea about the password checker is still the same. I was first amused by the fact that there is regex. I didn’t realize regex is used in text finding, and when I relate the expression to searching words in notepad, everything clears the way. As a matter of fact, I kind of like to use regex because it is interesting to match the characters without creating a for loop.

In Assignment 1, there were two parts I struggle. One was the regular expression. I was completely new to the concept, which took some time to get used to the syntax. The next one is exception classes. Even though I had learned exceptions in CMSC 203, I am still not comfortable with it because I don’t see how I can apply it until this assignment. Having exceptions or custom exceptions makes code more clear and concise. In the future, I might at all times add exceptions in my programs.

For the next project, which is about converting infix to postfix using stack and queue seems intimidating to me. Therefore, I will have to start early to read and practice more about what is Stack and Queue before to even start doing the project. It is important to know the basics before doing. For Assignment 1, I completed it in the hard way by trying every possible code to fix errors. It is not good practice for me.