

# Project 1

## Valid Password =

- ↳ 1+ numChar
- ↳ 1+ upAlChar
- ↳ 1+ lowAlChar
- ↳ 1+ specChar
- ↳ < 2 sameCharInSeq

## Strong Password =

- ↳  $\geq 10$  chars

## Weak Password =

- ↳  $6 \leq \text{length} \leq 9$
- ↳ must be valid

## Good Faith Attempt (GFA) requirements.

- ↳ Pass GFA\_test.java
- ↳ Pass test.java

## Expectations: as author

- ↳ include name on every java file
- ↳ a working "Check Password" button
- ↳ working "Check Passwords in file" button w/ feedback on which failed and why


## Requirements

- ↳ PasswordCheckerUtility class (PCU)
- ↳ exception classes for all exceptions in Java Doc
- ↳ check length first
- ↳ stop if password fails one rule
- ↳ method returns true if passes; if fails, it throws exception
- ↳ use regex for special chars
  - ↳ import Pattern + Matcher
- ↳ tests w/ minimum 4 cases
- ↳ capture actual v. expected results in write-up

## TIPS









- ↳ implement methods for PCU class
- ↳ create defined exceptions
- ↳ pass GFA tests + my own
- ↳ test GUI
- ↳ anticipate private tests

## To-do

 = done!

 = passes tests

- ↳ Exceptions (order matters)

-  ↳ Class LengthException (CLE)
-  ↳ Class NoUpperAlphaException (CNUAE)
-  ↳ Class NoLowerAlphaException (CNLAE)
-  ↳ Class NoDigitException (CNDE)
-  ↳ Class NoSpecialCharException (CNSCE)
-  ↳ Class InvalidSequenceException (CISE)
-  ↳ Class WeakPasswordException (CWPE)
-  ↳ Class UnmatchedException (CUE)

- ↳ PasswordCheckerUtility

- ✗  $\hookrightarrow$  sv compare Passwords (password, password Confirm)
  - $\hookrightarrow$  t CUE
- ✗  $\hookrightarrow$  sb compare Passwords With Return (pw, pw Confirm)
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  s ArrayList<String> get Invalid Passwords (passwords)
  - $\hookrightarrow$  r ArrayList of invalid Passwords
- ✗  $\hookrightarrow$  sb has Between Six And Nine Chrs (pw)
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb has Digit (pw)  $\rightarrow$  t CNDE
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb has Lower Alpha (pw)  $\rightarrow$  t CNLAE
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb has Same Char In Sequence (pw)  $\rightarrow$  t CISE
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb has Special Char (pw)  $\rightarrow$  t CNSCE
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb has Upper Alpha (pw)  $\rightarrow$  t CNUAE
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb isValidLength (pw)  $\rightarrow$  t CLF
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb isValidPassword (pw)  $\rightarrow$  t evero Exception
  - $\hookrightarrow$  r bool
- ✗  $\hookrightarrow$  sb isWeakPassword (pw)  $\rightarrow$  t CLPE
  - $\hookrightarrow$  r bool

## Logic Behind Methods

```

 $\hookrightarrow$  sv compare Passwords (String password, String password Confirm) {
    try {
        pw.equals(pwConfirm);
    } catch (UnmatchedException e) {
        throw new UnmatchedException();
    }
}

```