Objectives

- Process input and format output
- Implement sequential, repetition, and selection control structures
- Implement methods to organize code.
- Develop code that manipulate strings

Introduction

In this assignment, you will create a program where the user of your program attempts to guess a hidden phrase by guessing letters. A phrase consists of one or more related words, such as "frog" or "Pacific Ocean".

The structure of the game:

1. The computer is started with a phrase unknown to the player and then writes out a number of dashes equal to the characters in the phrase to the screen.
2. The human user (player) guesses letters. Whenever the player guesses a letter contained in the hidden phrase, the program reveals each instance of that letter in the word. Otherwise, the guess is wrong.
3. The game ends either when all the letters in the word have been guessed by the player correctly or when the player has run out of guesses.

To help guide you through the process, this assignment will be broken into pieces where one part builds on the next. You initially will create a program that displays an ASCII art guess scoreboard will be due. This program represents a significant milestone in the program and will be modified slightly in the final version of the assignment so that it correctly displays guesses in the complete assignment due the following week.

To help with your planning the overall structure and assessment points of the assignment are provided below.

- Assignment Part I: Guess Scoreboard Program (due first week): 40 points

- Assignment Part II: Completed Program (due second week): 50 points

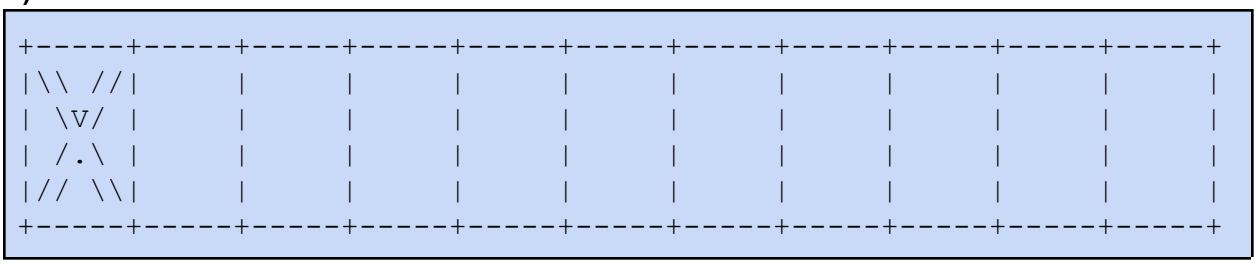
Assignment

Part I: Available Guess Scoreboard Program

In the first part of the assignment you will create an ASCII art piece representing a guess scoreboard for the game. The purpose is to visually display the number of guesses available and the total number of incorrect guesses so far. The guess board displays a box for each of the guesses available. If an incorrect guess is made, it is marked as an X in the first available box to the left.

The number of available guesses can vary --- ranging between 1 and 11 in this game. The scoreboard will display a different number of boxes dependent upon the number of guesses available. In this program, both the number of available guesses and the incorrect guesses can vary as inputs typed in by the user of the program. Examples of the display guest boxes for the range of guesses is shown below. You will need to match exactly the format of the guess board. You can assume that this program will always receive a valid input of total guesses and incorrect guesses

Figure: Display total guesses can be from 1 to 11 inclusive (with first box displaying an X)



Sample inputs and outputs for the program

```
Enter number of guesses allowed between 1 and 11
9
Enter number of incorrect guesses
7
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|\\ //|\\ //|\\ //|\\ //|\\ //|\\ //|\\ //|     |     |
| \V/ | \V/ | \V/ | \V/ | \V/ | \V/ | \V/ |     |     |
| /.\ | /.\ | /.\ | /.\ | /.\ | /.\ | /.\ |     |     |
|// \\|// \\|// \\|// \\|// \\|// \\|// \\|     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
Enter number of guesses allowed between 1 and 11
2
Enter number of incorrect guesses
0
+-----+-----+
|     |     |
|     |     |
|     |     |
|     |     |
+-----+-----+
```

```
Enter number of guesses allowed between 1 and 11
5
Enter number of incorrect guesses
3
+-----+-----+-----+-----+-----+
|\\ //|\\ //|\\ //|     |     |
| \V/ | \V/ | \V/ |     |     |
| /.\ | /.\ | /.\ |     |     |
|// \\|// \\|// \\|     |     |
+-----+-----+-----+-----+-----+
```

Requirements for Part I

- Create a class called **DrawScoreboard** and save it in a file called **DrawScoreboard.java**
- Create a method called **drawScoreboard** with the signature given below.
- In your **main** method, prompt the user for the number of guesses allowed and the number of incorrect guesses and save in an appropriate variable (NOTE: You need two variables and think about correct type and a good name. Use a Scanner object to read user input into the variables.)
- The **drawScoreboard** method should be invoked inside your **main** method using the number of guesses allowed and the number of incorrect guesses. You should then print the value returned from **drawScoreboard** to the console, with a newline following it.
- The result from **drawScoreboard** should provide the same number of boxes as total guesses horizontally in format shown in figures above.
- The result from **drawScoreboard** should contain an "X" in each box for each incorrect guess starting with the far left box and adding each X to the next available left box as shown in figures above.

drawScoreboard method

Method Signature

```
public static String drawScoreboard(int totalGuesses, int
incorrectGuesses)
```

- The purpose of this method is to provide a String representation of the current scoreboard when called.
- Takes two integers as parameters, total number of guesses and the number of incorrect guesses and returns a String to represent the scoreboard.

Submission for Part I

Submit your **DrawScoreboard.java** file to **Mimir** before the deadline. You may submit more than one time, the last attempt before the deadline is the one that is graded.

Before uploading your program, be sure to complete the following steps carefully:

- Your input and output should match the input and output test cases exactly, character for character.
- Make sure all your variables are declared at the beginning of the main method, before any other statements.
- Check that you have written the proper Javadoc @author and @version tags in the appropriate place.
- Run Checkstyle and eliminate ALL warnings about your code. (You should be doing this along the way and not just at the end of the assignment.)

Part II: Complete Assignment

This portion of the assignment implements the complete Guess the Phrase Game Assignment. You will need to create a new file called **PhraseGame.java** and copy over your drawScoreBoard method from Part I. After doing so, you no longer need **DrawScoreBoard.java** .

The main method in this program just provides an entry to kick off the game. It contains code to set the secret phrase and the total number of guesses and then calls the **playGame** method with those arguments. While a significant portion of the code for the game will be in the **playGame** method , it will also use the other three methods **displayGuessedLetters, displaySecretPhrase**, and **drawScoreboard** to implement the game.

Requirements for Part II

displayGuessedLetters method

```
public static String displayGuessedLetters(String letters)
```

- The purpose of this method is to provide an easily readable output of the letters that have been guessed.
- Method takes a string as an input parameter and returns the characters from the input string with some formatting applied

- The method returns a string containing the characters in the order they are in the string. It is not necessarily sorted.
- The guessedLetters parameters will be a string of lowercase letters (representing all guessed letters). There will be no spaces, digits, or other form of punctuation. The letters are not necessarily in sorted order, but will be unique. This means you will not see a lowercase letter more than once in the string.
- The return value should be a string containing a comma separated list of the lower case letters in the input parameter. Examples shown below

```
// For string input parameter "abfxz"
System.out.println(displayGuessedLetters("abfxz"));
// Results in the below being printed to screen
a, b, f, x, z

// For string input parameter "" or empty string


// For string input parameter "d"
d

// For string input parameter "xb"
x, b
```

displaySecretPhrase Method

```
public static String displaySecretPhrase(String secretPhrase,
                                         String guessedLetters)
```

- The purpose of this method is to provide a string that contains the “display” version of the secret phrase. The following rules apply for displaying the secret phrase:
    - For each letter that is not guessed, a dash is displayed
    - Spaces between words are displayed as spaces
    - astroph in words are displayed
    - For each guessed letter, the letter is displayed in the correct case. Meaning if it is capitalized in the secret phrase, then it is capitalized in the display secret phrase string.
- There are two input string parameters: the secret phrase, and the guessedLetters

- The secret phrase is the secret phrase which can consists of upper and lower case letters, apostrophes, and spaces. No other character will occur in the secret phrase.
- The guessedLetters parameter will meet the requirements described in the description above for displayGuessedLetters

- Examples shown below

```
// For the following statement
// System.out.println(displaySecretPhrase("Cat","xot"));
// Prints out
--t

// For the following statement
// System.out.println(displaySecretPhrase("cat","tcda"));
// Prints out
cat

// For the following statement
// System.out.println(displaySecretPhrase("You're great","yrte"));
// Prints out
Y--'re -re-t

// For the following statement
// System.out.println(displaySecretPhrase("Cats and dogs",""));
---- --- ----
// Prints out
```

playGame Method

```
public static void playGame(String secretPhrase, int totalGuesses)
```

- The playGame method will utilize a repetition structure to control the flow in the game. The game will proceed until an exit condition is reached.
- The method takes two parameters:
  - A string for the secret phrase
  - An integer for the number of guesses
- The game prompts the user to enter a character as a guess of a letter in the phrase. The input can either be an upper or lower case letter. However, it is consider a match if either the uppercase or lower case letter is in the phrase.
  - For example, entering the character 'A' (or 'a') at the prompt as a guess, matches all letters for **a** (either 'A' or 'a') in the phrase.

- The game performs the following decision logic after a character is entered by the player
    - If the character is a letter (upper or lower case) it will check to see if that letter has already been guessed. If it has not, it will store the letter guess in a string. If it has been, the user is prompted to enter another letter after a message indicating a duplicate. In either case, they are both considered as a valid guess and not stored as an incorrect guess
    - If the character is a letter but not in the secret phrase, a message indicating an incorrect guess has been made
    - If an '!' is entered, the game ends immediately
    - If any other character is entered, a message is displayed indicating that it is not a valid selection. This also does not result in an incorrect guess
- The game ends either by:
    - Entering the '!' as a guess
    - Guessing all of the letters in the phrase or a WIN for the player
    - Exceeding the number of guesses before guessing all letters in the phrase. A LOSE for the player.
- The program sample outputs demonstrate the flow of execution and the expected output. NOTE: your output must match EXACTLY the output shown in the examples to receive credit in the assignment.

You will submit your **PhraseGame.java** file. The main method will not be tested, so you can use the **main** method for testing the other methods using some of the example invocations and print statements given above.

Win Example

```
//playGame("Cat",2);

+-----+-----+
|     |     |
|     |     |
|     |     |
|     |     |
+-----+-----+
Secret Phrase:   ---
Guessed Letters:
Enter a letter to guess or "!" to end the game
d
No, d is not in the phrase

+-----+-----+
|\\ //|     |
| \V/ |     |
| /.\ |     |
|// \\|     |
+-----+-----+
Secret Phrase:   ---
Guessed Letters: d
Enter a letter to guess or "!" to end the game
c

+-----+-----+
|\\ //|     |
| \V/ |     |
| /.\ |     |
|// \\|     |
+-----+-----+
Secret Phrase:   C--
Guessed Letters: d, c
Enter a letter to guess or "!" to end the game
t

+-----+-----+
|\\ //|     |
| \V/ |     |
| /.\ |     |
|// \\|     |
+-----+-----+
Secret Phrase:   C-t
Guessed Letters: d, c, t
Enter a letter to guess or "!" to end the game
a

=======================================
```

```
You won!
The phrase was "Cat"
=====================================
```

Lose Example

```
//playGame("Cat Dog",3);

+-----+-----+-----+
|     |     |     |
|     |     |     |
|     |     |     |
|     |     |     |
+-----+-----+-----+
Secret Phrase:   --- ---
Guessed Letters:
Enter a letter to guess or "!" to end the game
d

+-----+-----+-----+
|     |     |     |
|     |     |     |
|     |     |     |
|     |     |     |
+-----+-----+-----+
Secret Phrase:   --- D--
Guessed Letters: d
Enter a letter to guess or "!" to end the game
m
No, m is not in the phrase

+-----+-----+-----+
|\\ //|     |     |
| \V/ |     |     |
| /.\ |     |     |
|// \\|     |     |
+-----+-----+-----+
Secret Phrase:   --- D--
Guessed Letters: d, m
Enter a letter to guess or "!" to end the game
o
```

```
+-----+-----+-----+
|\\ //|     |     |
| \V/ |     |     |
| /.\ |     |     |
|// \\|     |     |
+-----+-----+-----+
Secret Phrase:   --- Do-
Guessed Letters: d, m, o
Enter a letter to guess or "!" to end the game
x
No, x is not in the phrase

+-----+-----+-----+
|\\ //|\\ //|     |
| \V/ | \V/ |     |
| /.\ | /.\ |     |
|// \\|// \\|     |
+-----+-----+-----+
Secret Phrase:   --- Do-
Guessed Letters: d, m, o, x
Enter a letter to guess or "!" to end the game
c

+-----+-----+-----+
|\\ //|\\ //|     |
| \V/ | \V/ |     |
| /.\ | /.\ |     |
|// \\|// \\|     |
+-----+-----+-----+
Secret Phrase:   C-- Do-
Guessed Letters: d, m, o, x, c
Enter a letter to guess or "!" to end the game
y
No, y is not in the phrase

+-----+-----+-----+
|\\ //|\\ //|\\ //|
| \V/ | \V/ | \V/ |
| /.\ | /.\ | /.\ |
|// \\|// \\|// \\|
+-----+-----+-----+

=====================================
No more guesses left. Game over!
The phrase was "Cat Dog"
=====================================
```

Other functionality example

```
//playGame("Hamburger and fries",5);

+-----+-----+-----+-----+-----+
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
+-----+-----+-----+-----+-----+
Secret Phrase:   --------- --- -----
Guessed Letters:
Enter a letter to guess or "!" to end the game
H

+-----+-----+-----+-----+-----+
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
+-----+-----+-----+-----+-----+
Secret Phrase:   H-------- --- -----
Guessed Letters: h
Enter a letter to guess or "!" to end the game
A

+-----+-----+-----+-----+-----+
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
+-----+-----+-----+-----+-----+
Secret Phrase:   Ha------- a-- -----
Guessed Letters: h, a
Enter a letter to guess or "!" to end the game
3
Not a valid guess!
+-----+-----+-----+-----+-----+
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
+-----+-----+-----+-----+-----+
Secret Phrase:   Ha------- a-- -----
Guessed Letters: h, a
Enter a letter to guess or "!" to end the game
(
Not a valid guess!
```

```
+-----+-----+-----+-----+-----+
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
+-----+-----+-----+-----+-----+
Secret Phrase:   Ha------- a-- -----
Guessed Letters: h, a
Enter a letter to guess or "!" to end the game
A
You already guessed a

+-----+-----+-----+-----+-----+
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
+-----+-----+-----+-----+-----+
Secret Phrase:   Ha------- a-- -----
Guessed Letters: h, a
Enter a letter to guess or "!" to end the game
a

You already guessed a
+-----+-----+-----+-----+-----+
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
+-----+-----+-----+-----+-----+
Secret Phrase:   Ha------- a-- -----
Guessed Letters: h, a
Enter a letter to guess or "!" to end the game
!

=====================================
Game ended early
=====================================
```

Submission for Part II

Submit your **PhraseGame.java** file to **Mimir** before the deadline. You may submit more than once.

Before uploading your program, be sure to complete the following steps carefully:

- Your input and output should match the input and output test cases exactly, character for character.
- Check that you have written the proper Javadoc @author and @version tags in the appropriate place.
- Run Checkstyle and eliminate ALL warnings about your code. (You should be doing this along the way and not just at the end of the assignment.)