Workshop 4

Name: Daryan Chan

ID: 113973192

Section: NAA

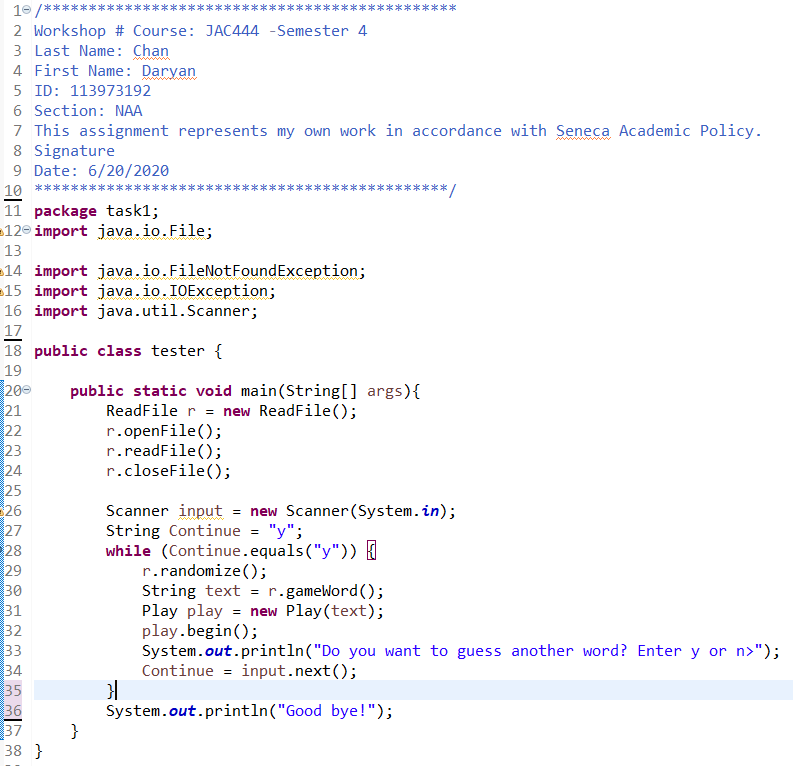


Figure tester class.

This was the last class I coded for task 1 and it basically utilize the other 2 classes. The first part was opening the file containing the game words. To begin, I instantiated the ReadFile class. I did this to call the openFile, readFile, and closeFile methods. Their functions will be explained in the second page.

After the file was been extracted and closed, I created a Scanner object that will be used later when the program ask the user if he/ she wants to continue playing. I created a while loop that will repeat as long as Continue string variable is “y”. As long as it’s y, the methods inside the loop will run in sequence. If the user enters “n”, then the program will end.

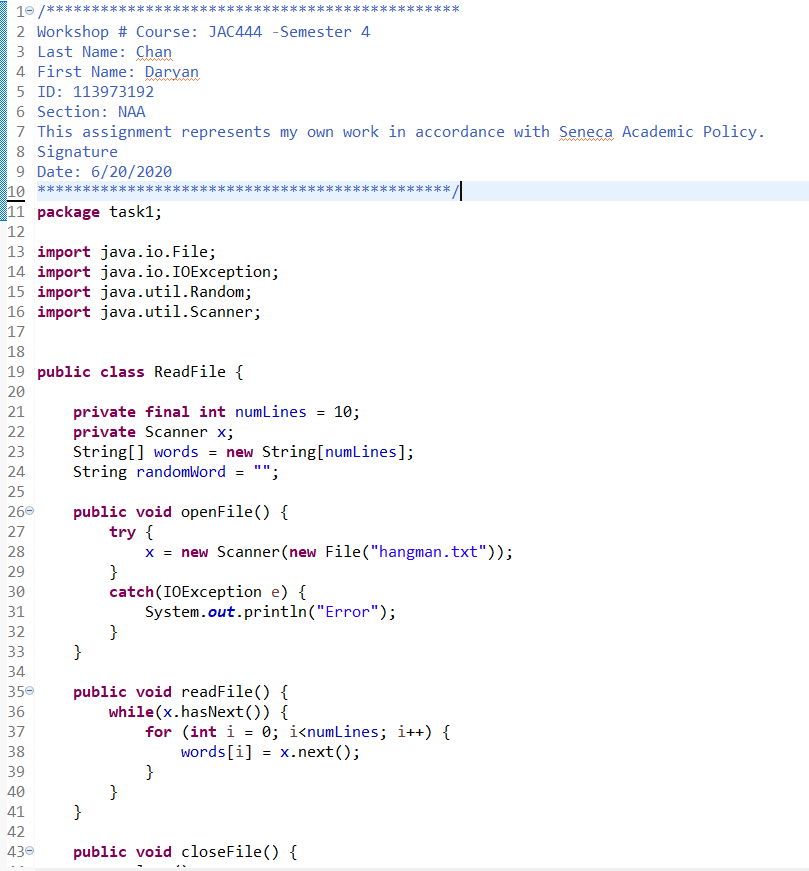


Figure ReadFile class and its methods and instance variables

The first thing I did in this class was creating the variables. Since I know how many lines there were in the text file, I created numLines to store the number of lines. The reason being is that each line will contain a word so that it will be easier to randomly choose a word for the game. After I created a Scanner x variable so I can use the Scanner feature easier and for each methods to be able to utilize it. I then created an array of words that match the number of lines and an empty string that will later contain the random word.

The first method basically opens the file. I did this using the try and catch method. The logic behind this is the file will always be open. If somehow the file is missing, it will catch the exception and print out an error message.

The next method is to read the file, readFile(). The purpose of this function is to go down line by line, copying each word into the words array.

The next method basically closes the file once its purpose is finished.

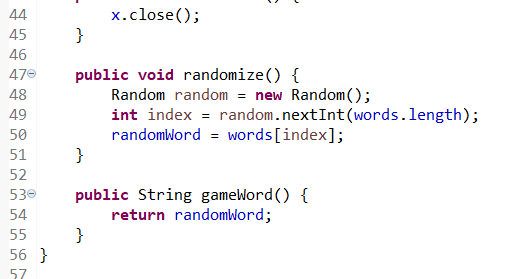


Figure ReadFile class continuation

The randomize method is self explanatory. It basically uses the Random class and generates a random index number from the words array. Then the number is used to pick which word from the array to use and stores that in the randomWord variable.

The last method returns back the randomWord.

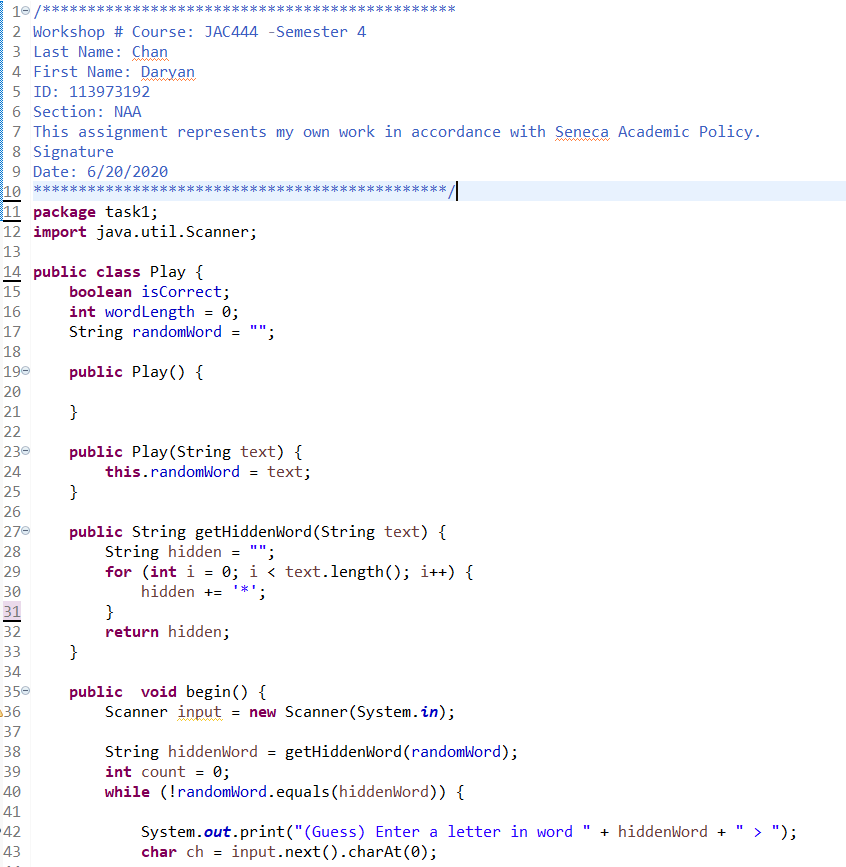


Figure Play class and its instance variables and methods

The first thing I did was created the constructors. The one argument constructor is used to initialize the randomWord string variable, which will be the word we will use for the game. The getHiddenWord method accepts one string, which is the game word. The purpose of this method is to hide the word. Therefore, it will determine the length of the game word and basically hidden with asterisk.

The next method is the core of this program. It will go through a while loop as long as the hidden word doesn’t match the hiddenWord with the asterisk. As long as this loop runs, it will prompt the user to take a guess. Once the user has entered a character, it will be stored in ch variable.

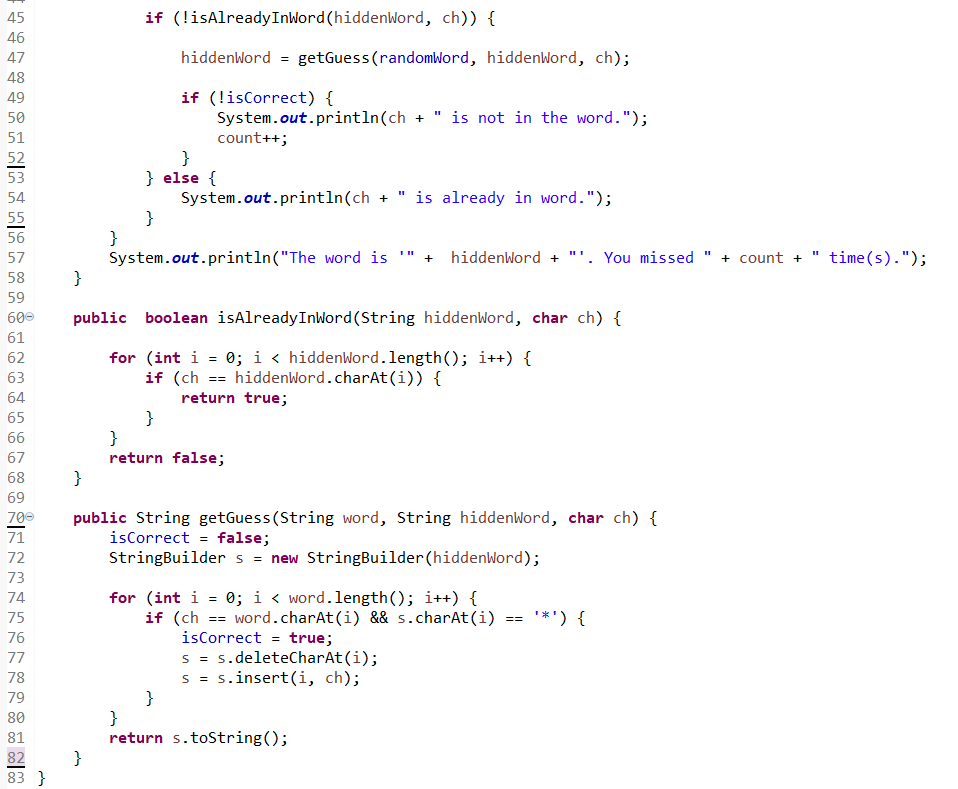


Figure Play class continuation

Once the character from the user has been entered, it goes through an if condition to determine if that character is already in the word. In other words, it’s seeing if the user has already entered that character. That is the purpose of the Boolean method below the begin method. The logic behind this is to take that character the user entered and see if it matches any characters in the hiddenWord using for loop. It will return true or false depending on what it finds.

If the word is not already in the hiddenWord, it calls the next method, getGuess. This method accepts 3 parameters, hiddenWord, randomWord and the ch. Using these three data, it undergoes a for loop based on the length of the randomWord and determines if the character matches any character in the randomWord. If it does, it also makes sure that the hidden word has an asterisk in the same index position. Once all these conditions have been met, it deletes the asterisk at that index position and replaces it with the character the user entered. Once that is done, this method returns the modified hiddenWord.

If the character is wrong, the counter will be increased for each wrong guess.

Once the while loop condition has been fulfilled, it will call the last code in the method and display what the word is and the number of times you guessed it wrong.

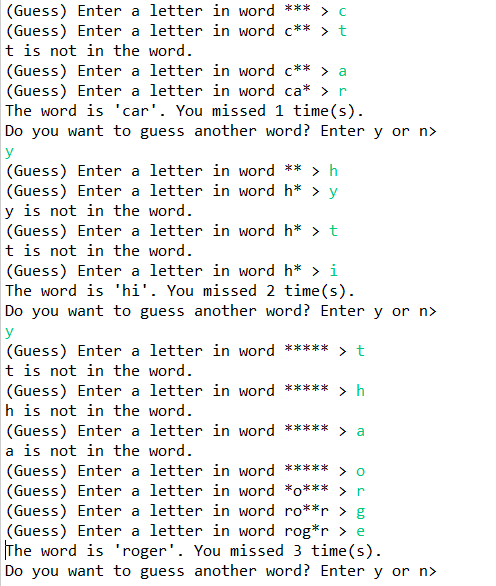


Figure task 1 results

Task 2

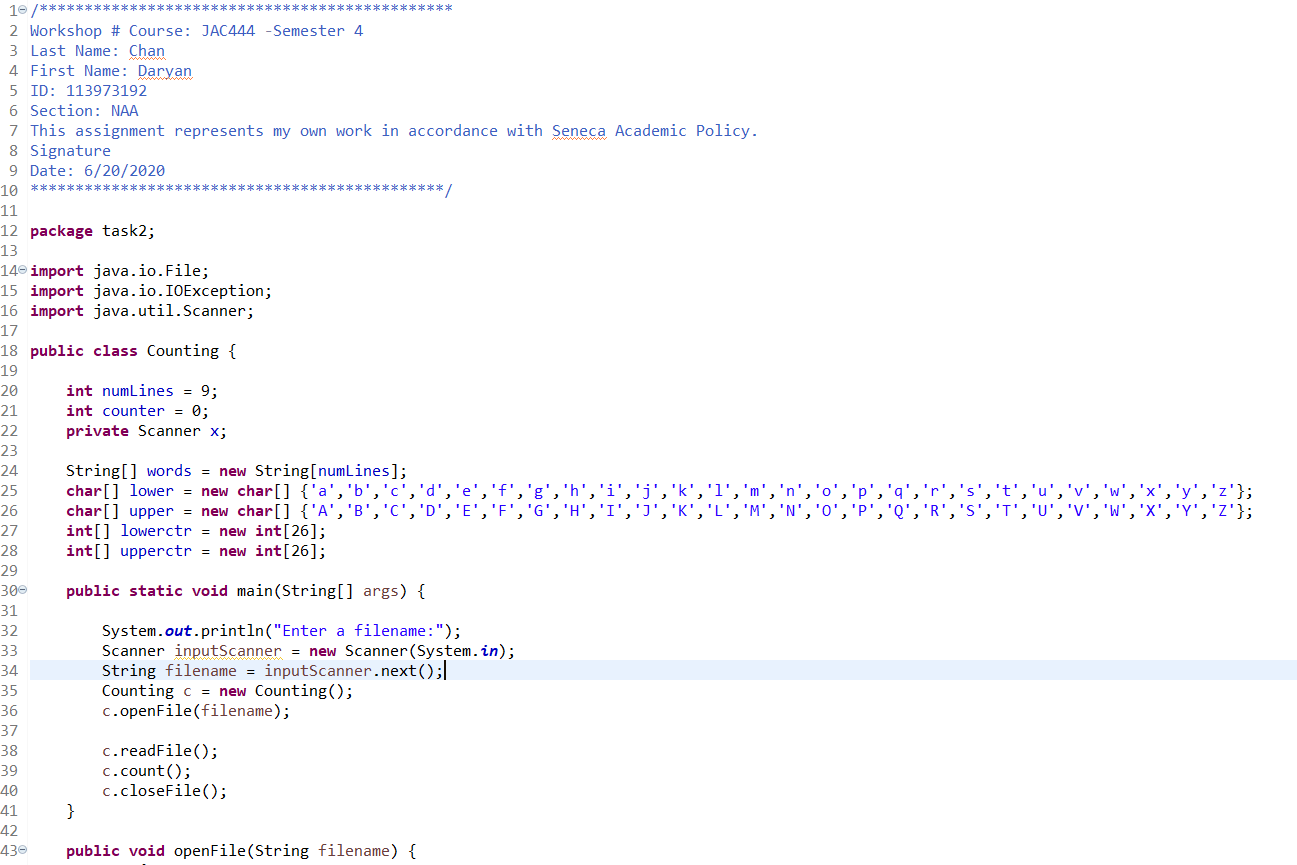


Figure Counting class and its methods and instance variables

This task takes what is used from task 1 and applies it to this task too. The first 3 instance variables has the same concepts as the previous task. The new thing I added are the arrays. The lower and upper array contains all the alphabets and the purpose of this is to match every alphabet in the file. The lowerctr and upperctr basically counts how many times each letter occurs.

The first thing this program does is it prompts the user to enter the filename. The name of the file is stored in the string filename variable. The Counting class is then instantiated so that I can call up its methods. The first method called is the openFile method and it takes the filename as its parameter.

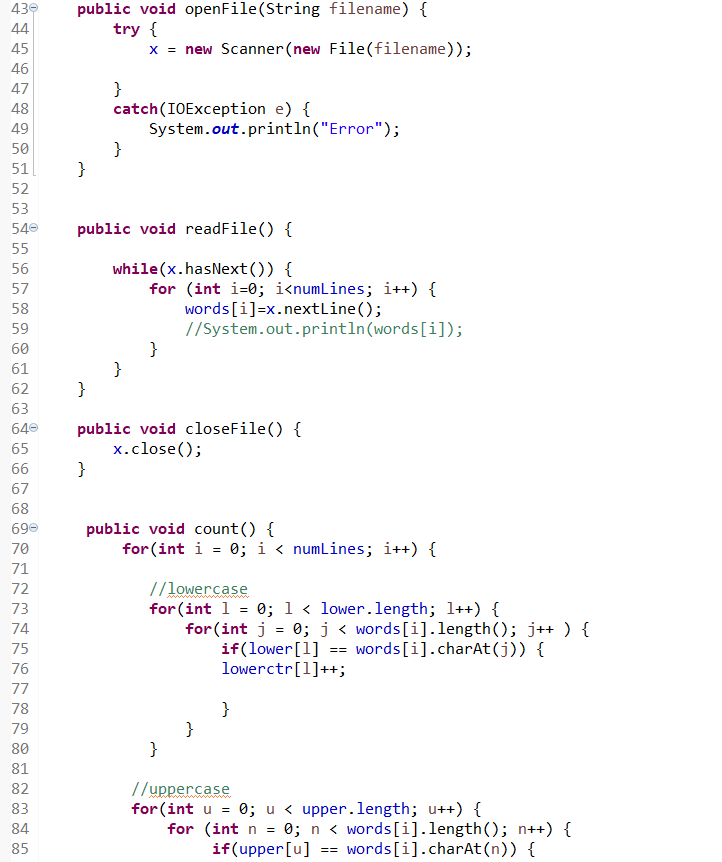


Figure Counting class continuation

The openFile takes the filename and opens it in the try catch method. If there is an error with opening the file, the catch will catch the exception and prints out an error message.

The next method basically reads the file. It goes through a while loop and as long as it detects more data within the file, it will go through a for loop and copies the content into the words array.

The next method basically closes the file when it’s done using it.

The count method goes through each string word variable and runs it along the upper and lower array. If there is a match for any case specific alphabets, the alphabet counters will record it.

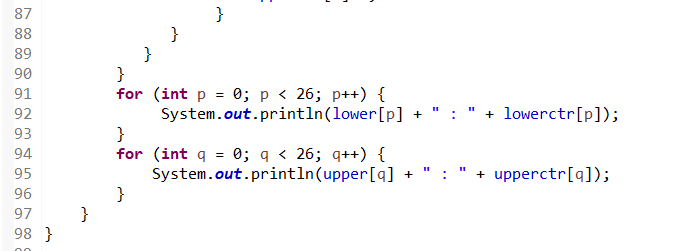


Figure Counting class continuation

The last lines of codes will then print out each alphabet and its recorded number of occurrence.

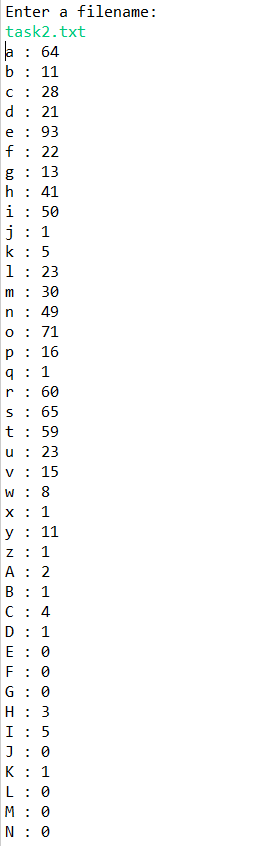
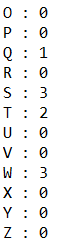


Figure results

This is the result of the program.

**The file content used in task 2**

By Sara Harrison Science

Ask the Know-It-Alls: What Is the Coronavirus?

Q What is the coronavirus?

Coronaviruses are a family of hundreds of viruses that can cause fever, respiratory problems, and sometimes gastrointestinal symptoms too. The 2019 novel coronavirus is one of seven members of this family known to infect humans, and the third in the past three decades to jump from animals to humans. Since emerging in China in December, this new coronavirus has caused a global health emergency.

How does it spread?

Its likely to be transmitted in droplets from coughing or sneezes, and the virus has a two- to 14-day incubation period. That means people could be infectious for quite a while before symptoms like fever, cough, or shortness of breath emerge.

What are the particular symptoms of Covid-19?

In the confirmed cases so far, most people get a fever with a dry cough; smaller numbers of folks might experience shortness of breath, a sore throat, or a headache.

How can I avoid catching the coronavirus?