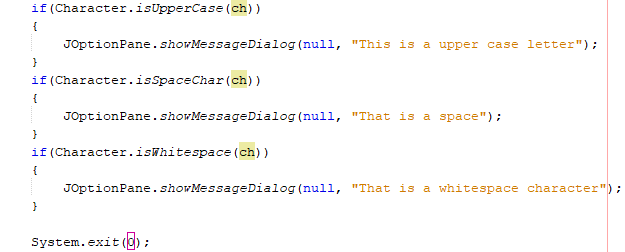
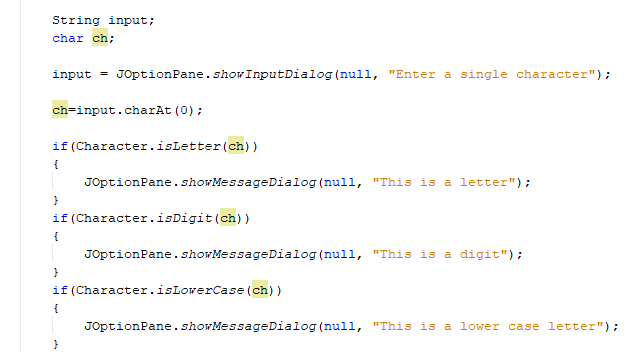
Objectives:

* Character Testing, String methods & GUI Apps, String Builder, List Arrays, Sub Strings, File, File Writer, Buffer Writer and Readers, String methods & GUI Apps

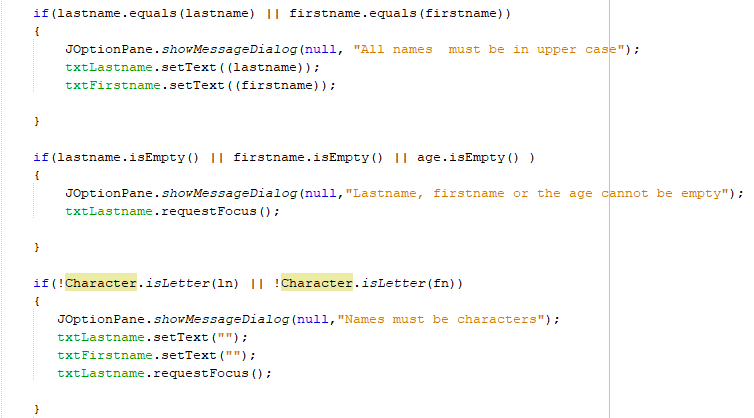
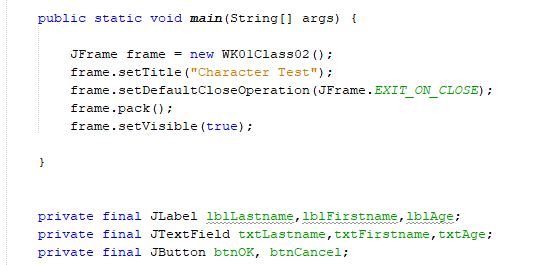
**There are 7 projects on this document each worth 14.2%**

The **Character** class is a wrapper class for the **char** data type. It provides numerous methods for testing and converting character data.

1. Create a class and name it **WK01Class01** and type in the following code, be sure to create the main method, you can use a shortcut *psvm* then press the tab button, this will automatically create the main method.



1. This class example will show certain validations for text fields using the. **toUpperCase,. IsEmpty**, **.isLetter** methods. Create a class and name it **WK01Class02** and type in the following code:

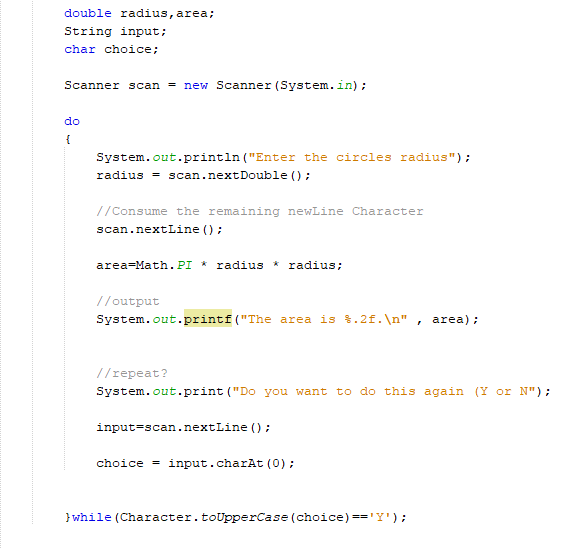


The JFrame must equal the name of the class and one important method is the setDefaultCloseOperation which will close every operation that you execute within the frame.

The ! symbol states if the Character is NOT a letter then execute what’s between the curly braces.

The number format exception class allows only numbers to be entered a text field

1. Create a class and name it **WK01Class03** and type in the following code:



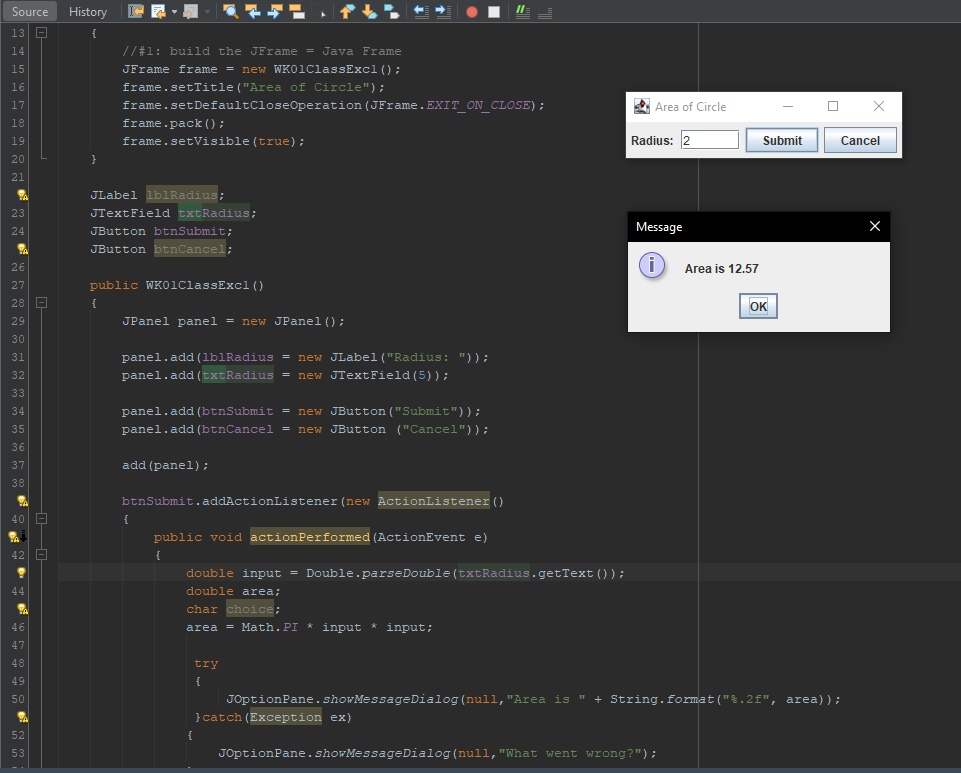
Do this, while that.

The Do while will always execute the code in curly brackets and will ask the user if to proceed or NOT.

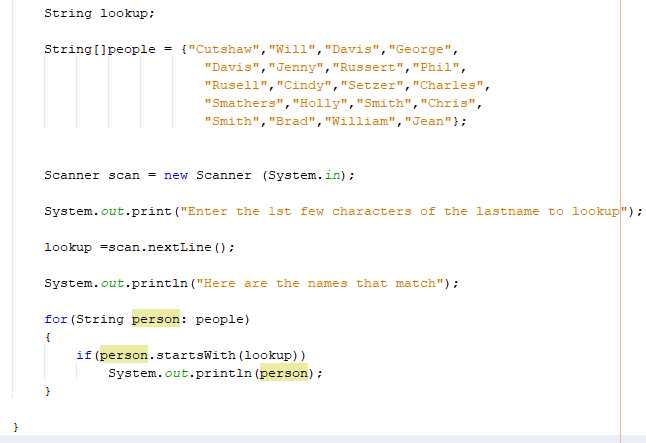
**Project #1:**

Create a new class and name it **WK01ClassExc1** Convert #3 into a GUI application that will perform the same algorithm

**#1 Print screen the running application with code below here**

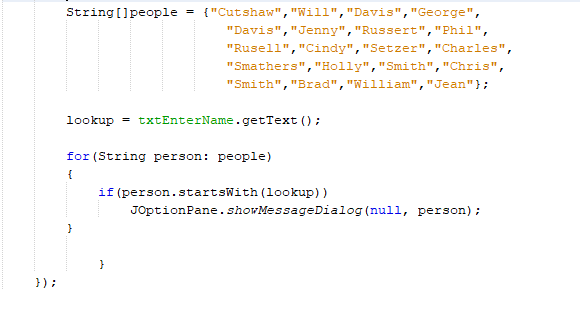
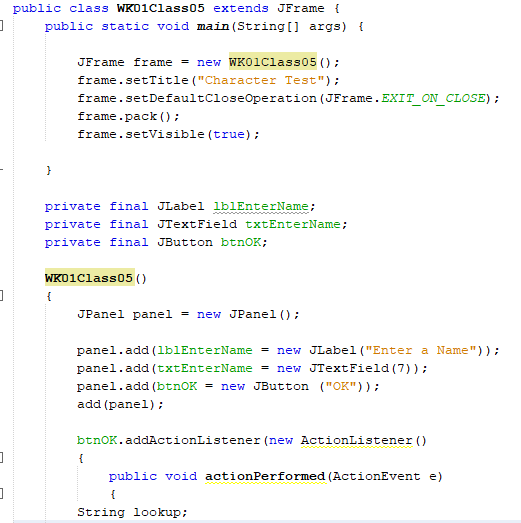
****

1. Create a class and name it **WK01Class04** and type the following code below:



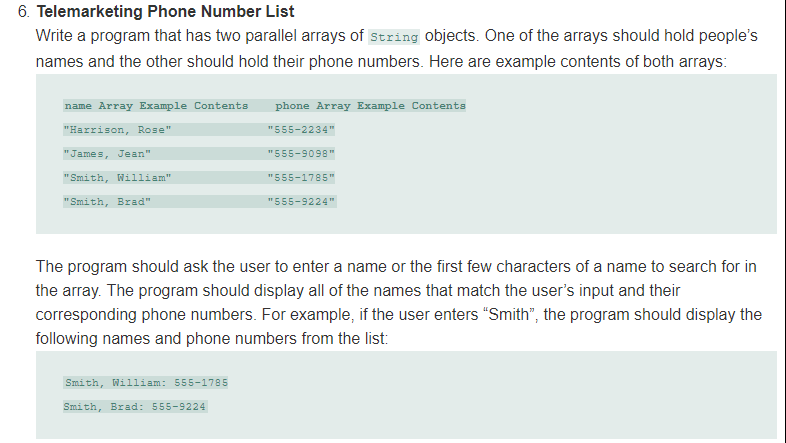
The people variable is an array variable that can hold multiple values

1. Converting #4 into a GUI interface, create a class and name it **WK01Class05** and type in the following code below on the next page.

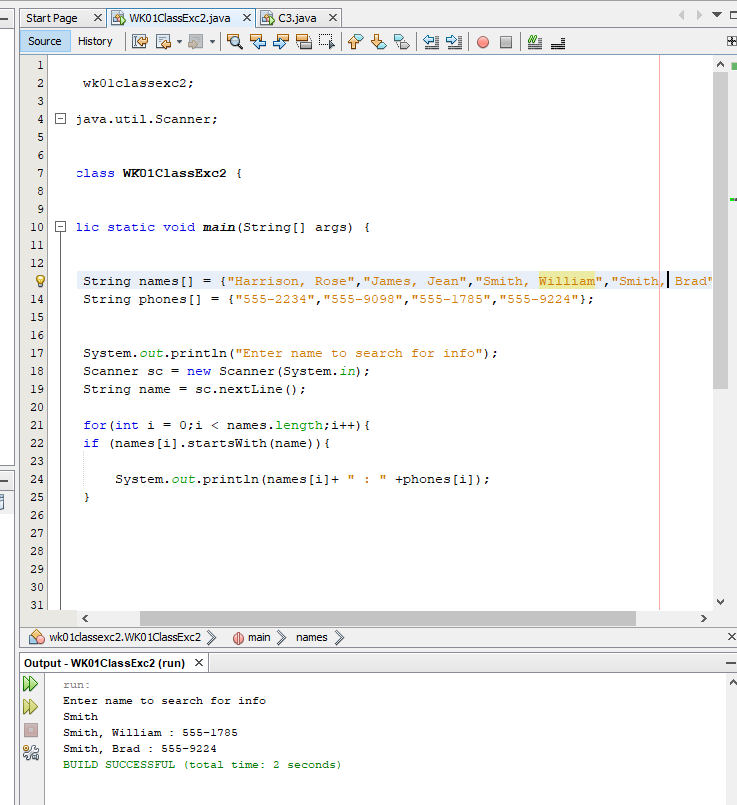


**Project #2:**

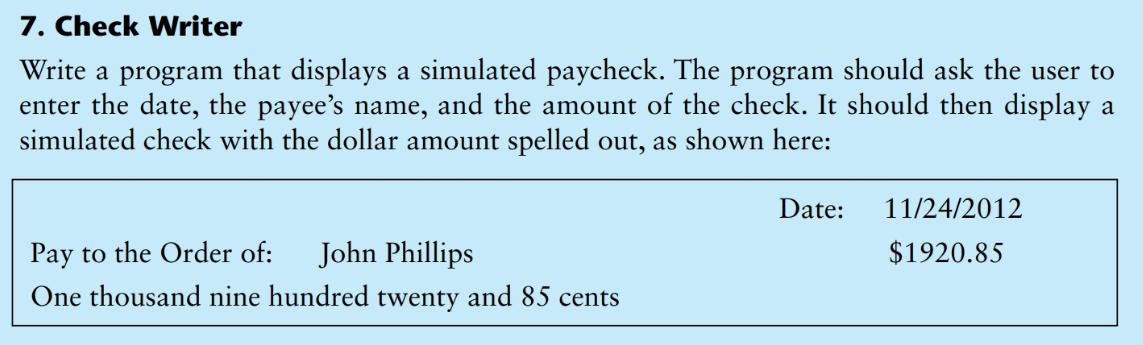
Create a new class and name it **WK01ClassExc2** and solve/complete the problem below:



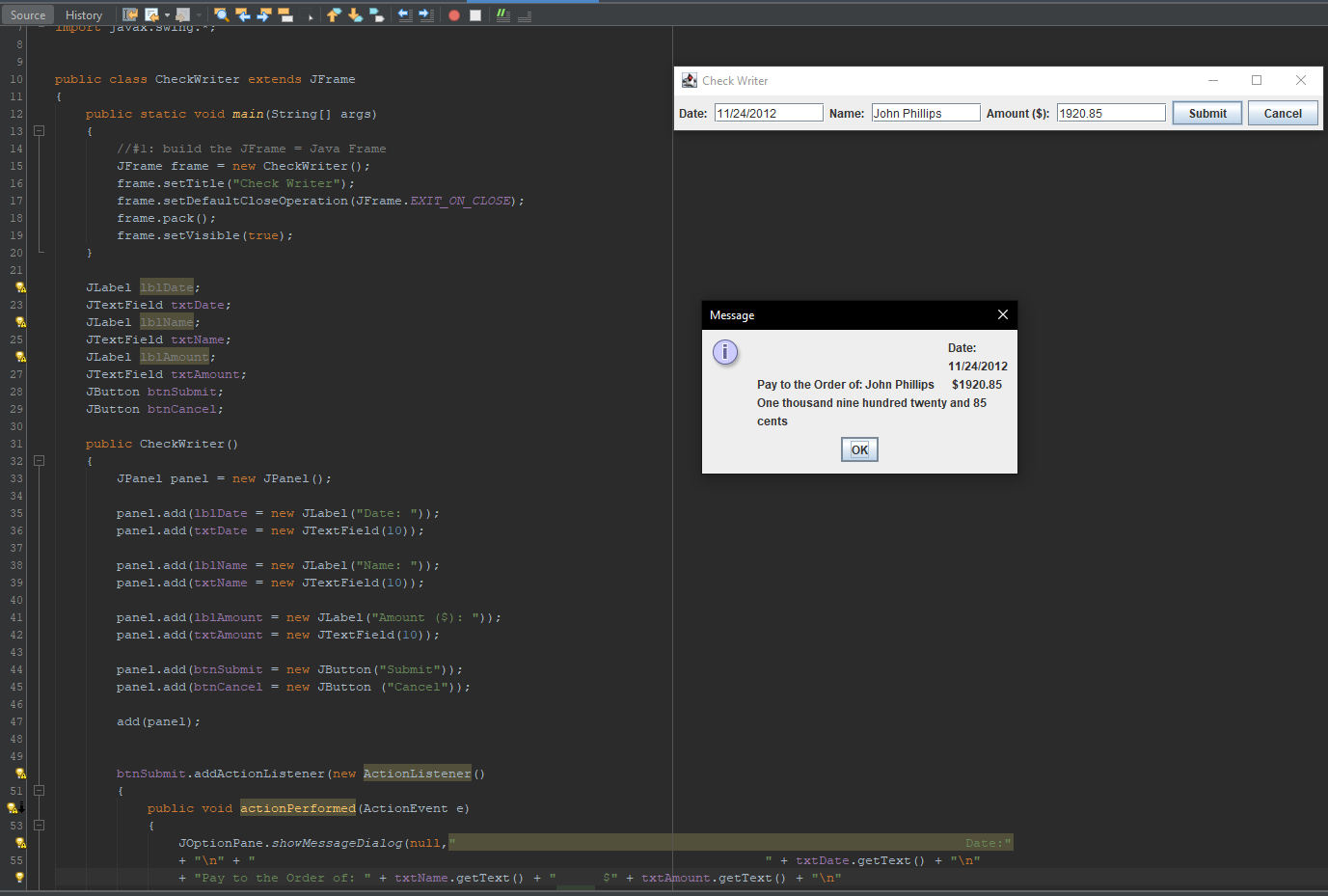
**#2 Print screen the running application with code below here**



**Project #3: Create a JFrame based on the following below.**

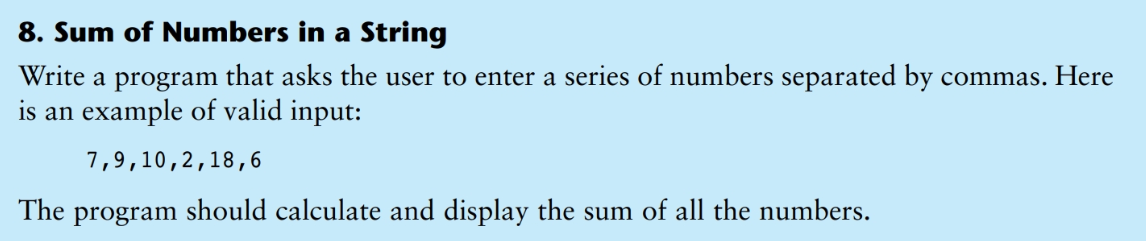


**#3 Print screen the running application with code below here**

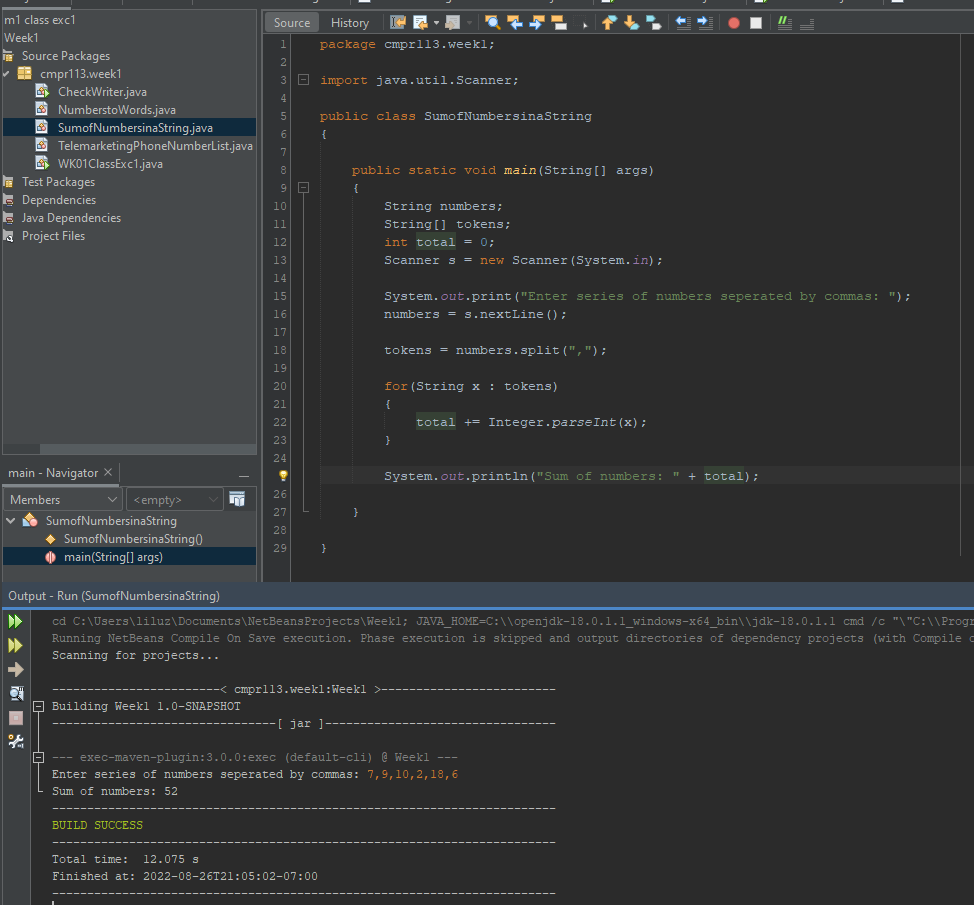


**Project #4:**

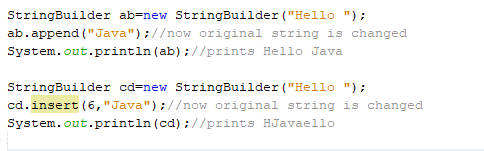
Note: use the 6 numbers as shown below



**#4 Print screen the running application with code below here**



1. Create a class and name it **WK02Class01** and type in the following code



Output:



H E L L O

1 2 3 4 5 6 >> insert 6 characters the 6th space equals the empty space

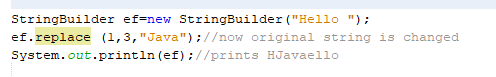
1. Change the number 6 for the insert to 7 and run the application, notice that we get an error because there is NO 7th string



1. Change the number to 1 and run the application, notice it retrieves the first character and after the word Java retrieves the rest



1. The StringBuilder replace() method replaces the given string from the specified beginIndex and endIndex.



Output



1. The delete() method of StringBuilder class deletes the string from the specified beginIndex to endIndex.

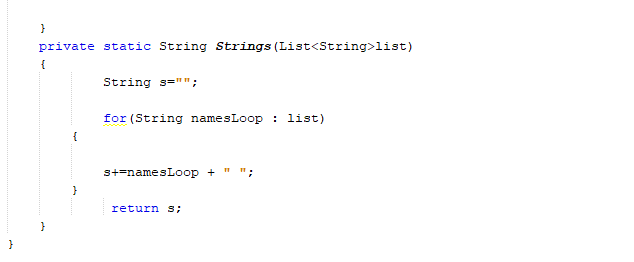


Output



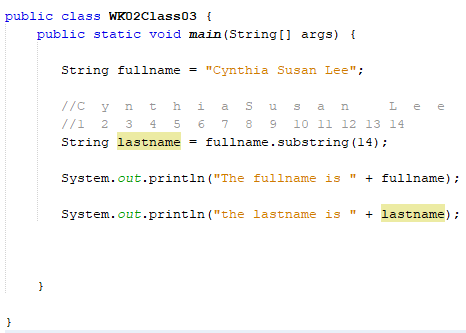
1. Create a class and name it **WK02Class02** and type in the following code, See next page….

The difference between Strings and StringBuilder is that Strings are **immutable** which means once they are created they cannot be changed. You can change the contents of a StringBuilder.



Substring method returns a copy of a substring from the calling object.

1. Create a class and name it **WK02Class03** and type in the following code



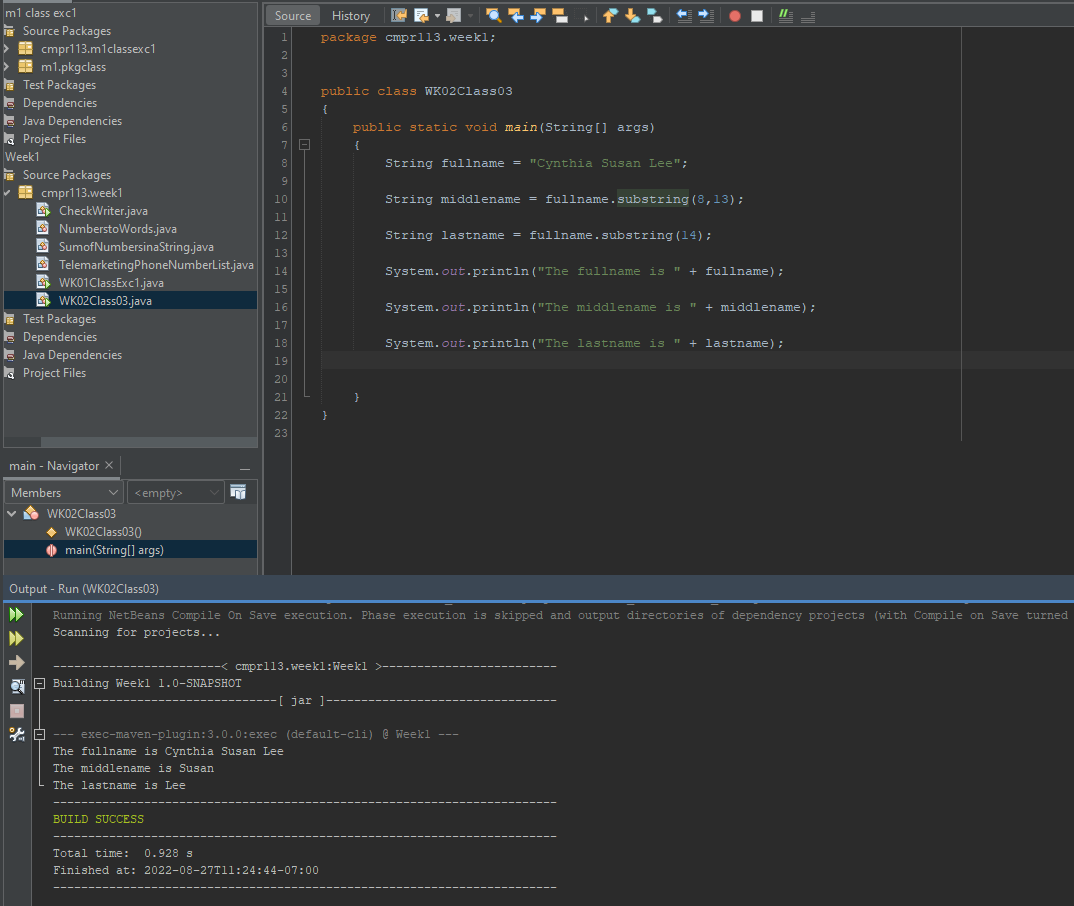
Output



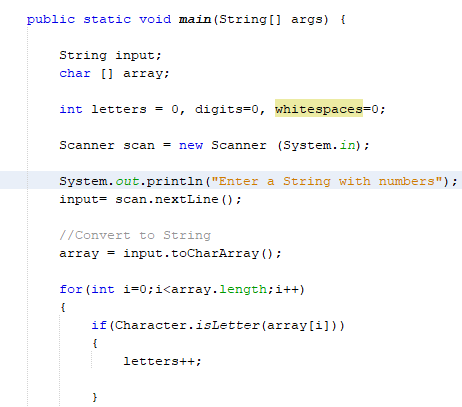
**Project #5:**

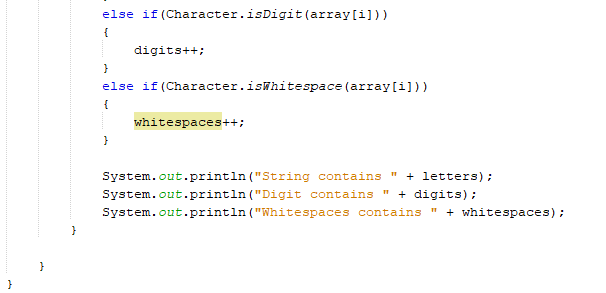
Modify the code above (#7) to return the middlename **Susan**

**#5 Print screen the running application with code below here**



1. Create a class and name it **WK02Class04** and type in the following code on the next page…

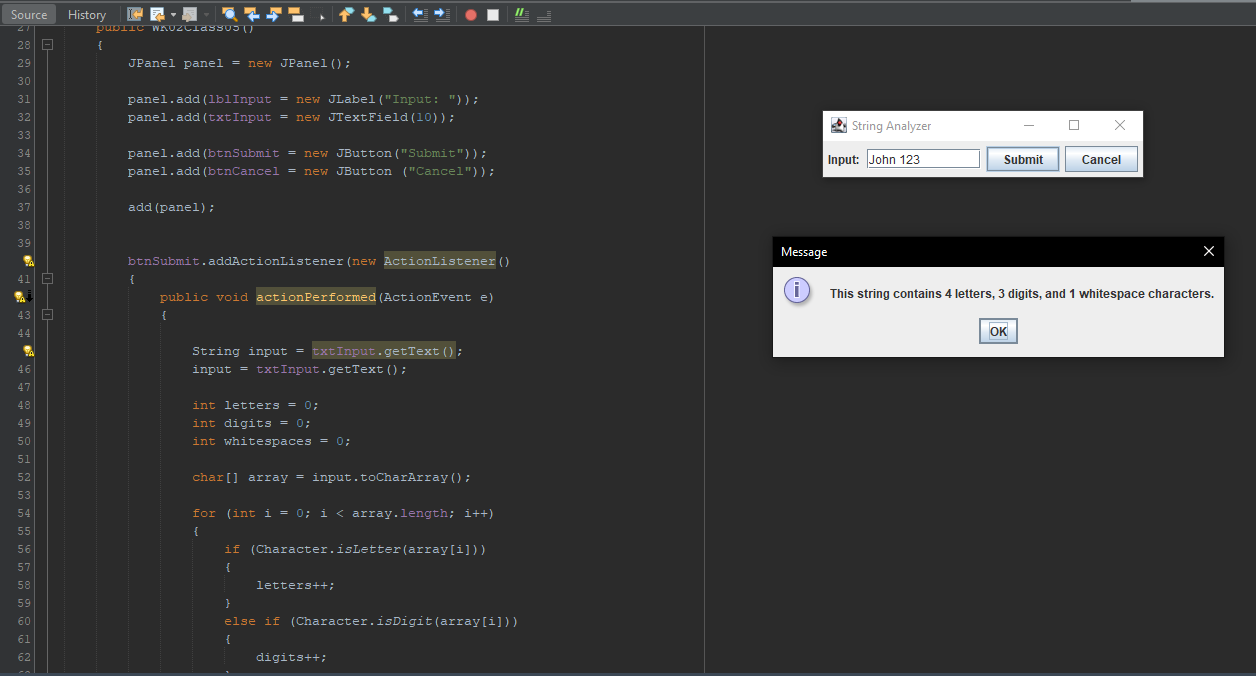




**Project #6**

1. Create a class and name it **WK02Class05** and create a GUI application for #8, print screen below where indicated.

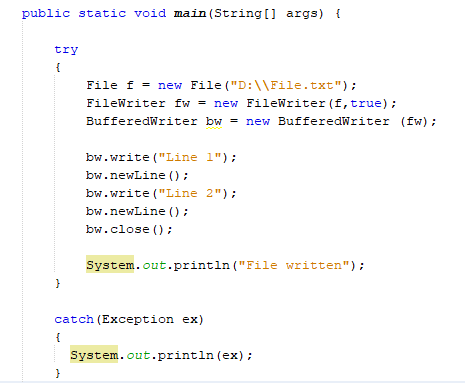
**#6 Print screen the running application with code below here**



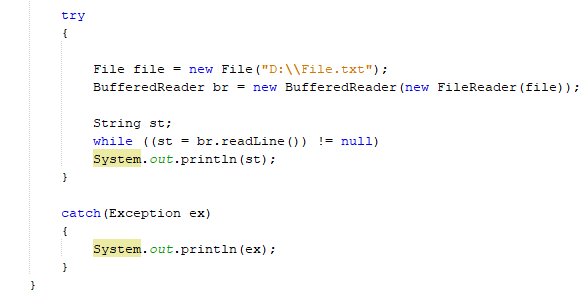
Class Exercise #06 will focus on reading and writing data from and too a text file

1. Create a class and name it **WK02Class06** and type in the following code below:

Writing to a file



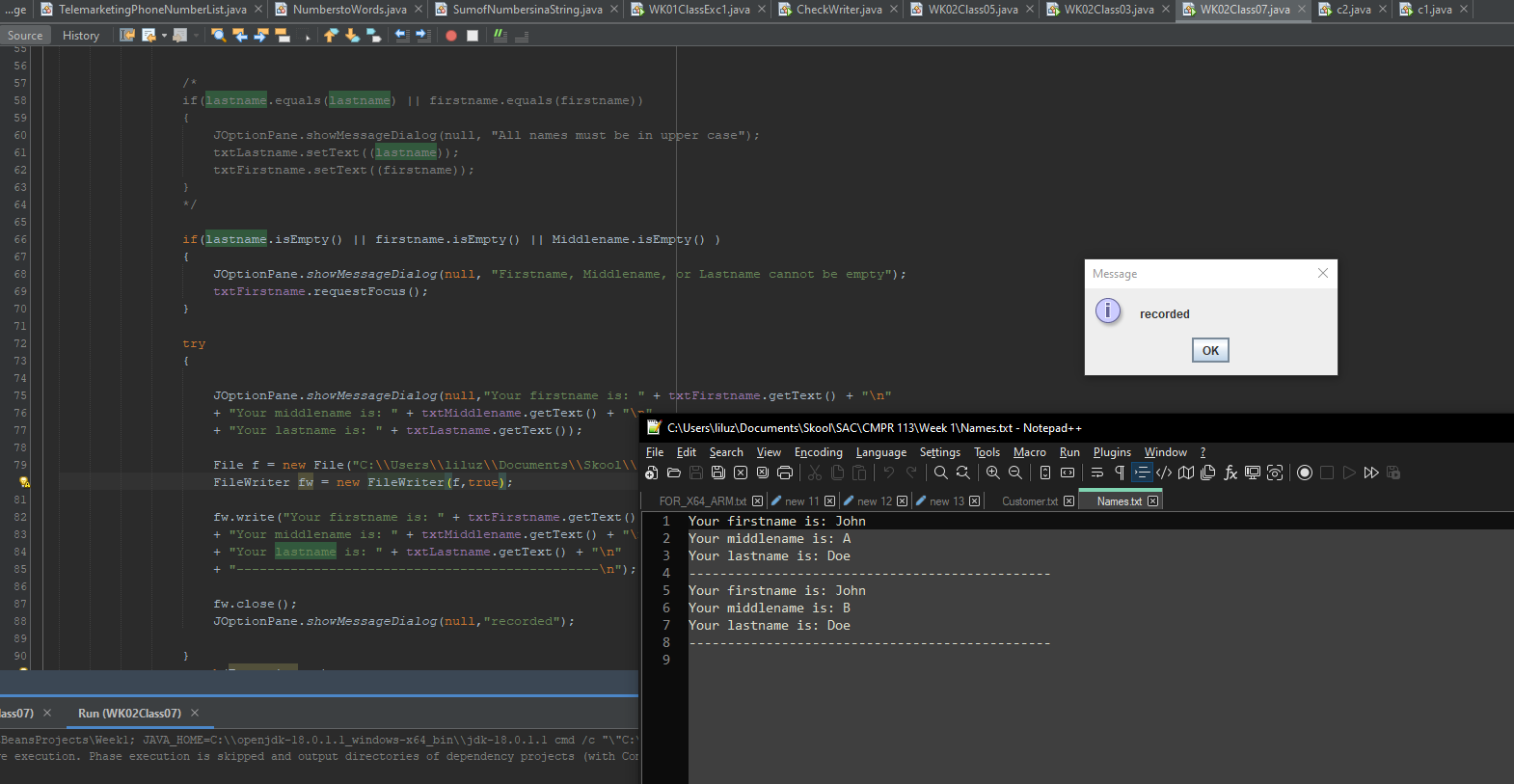
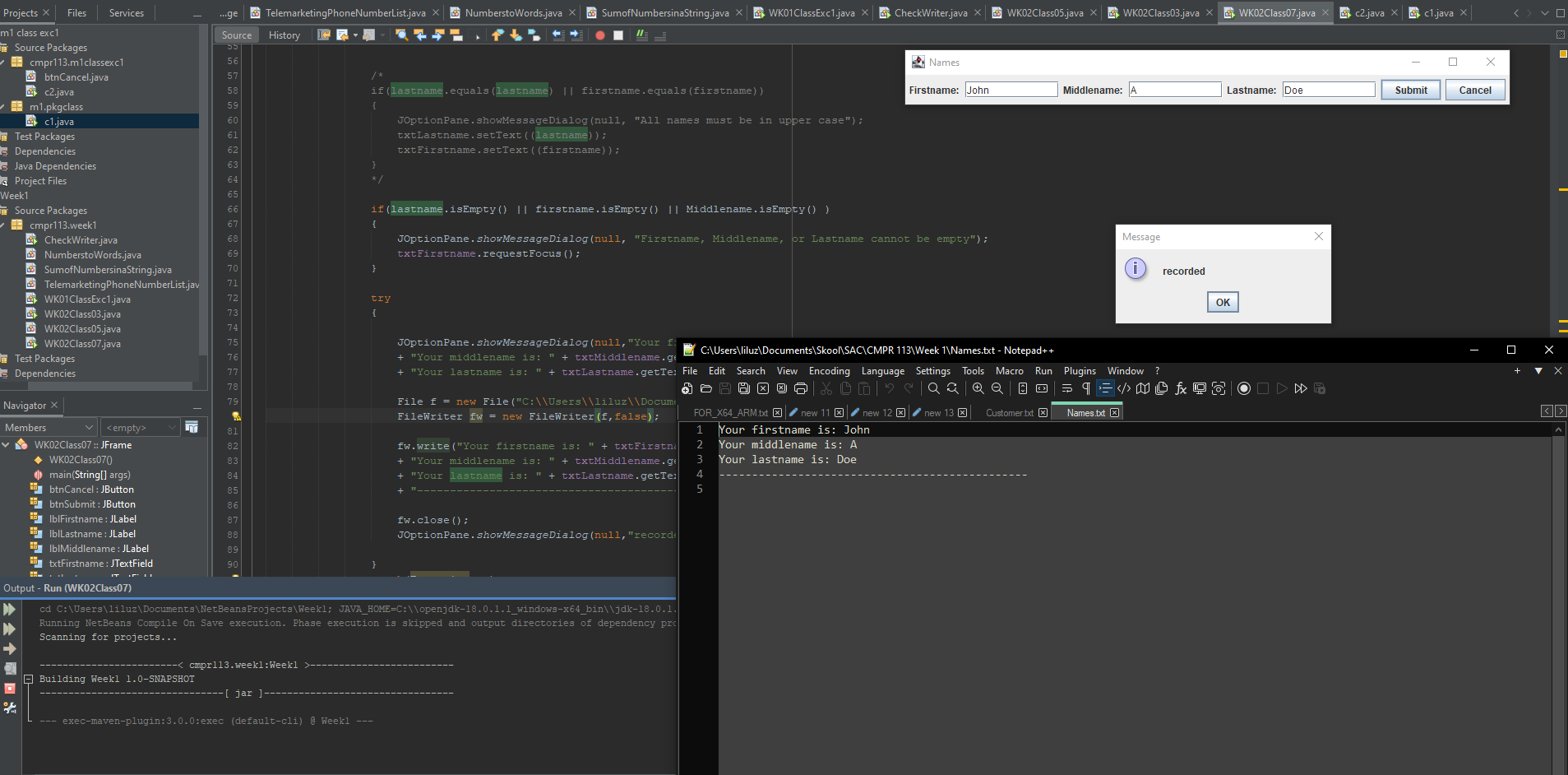
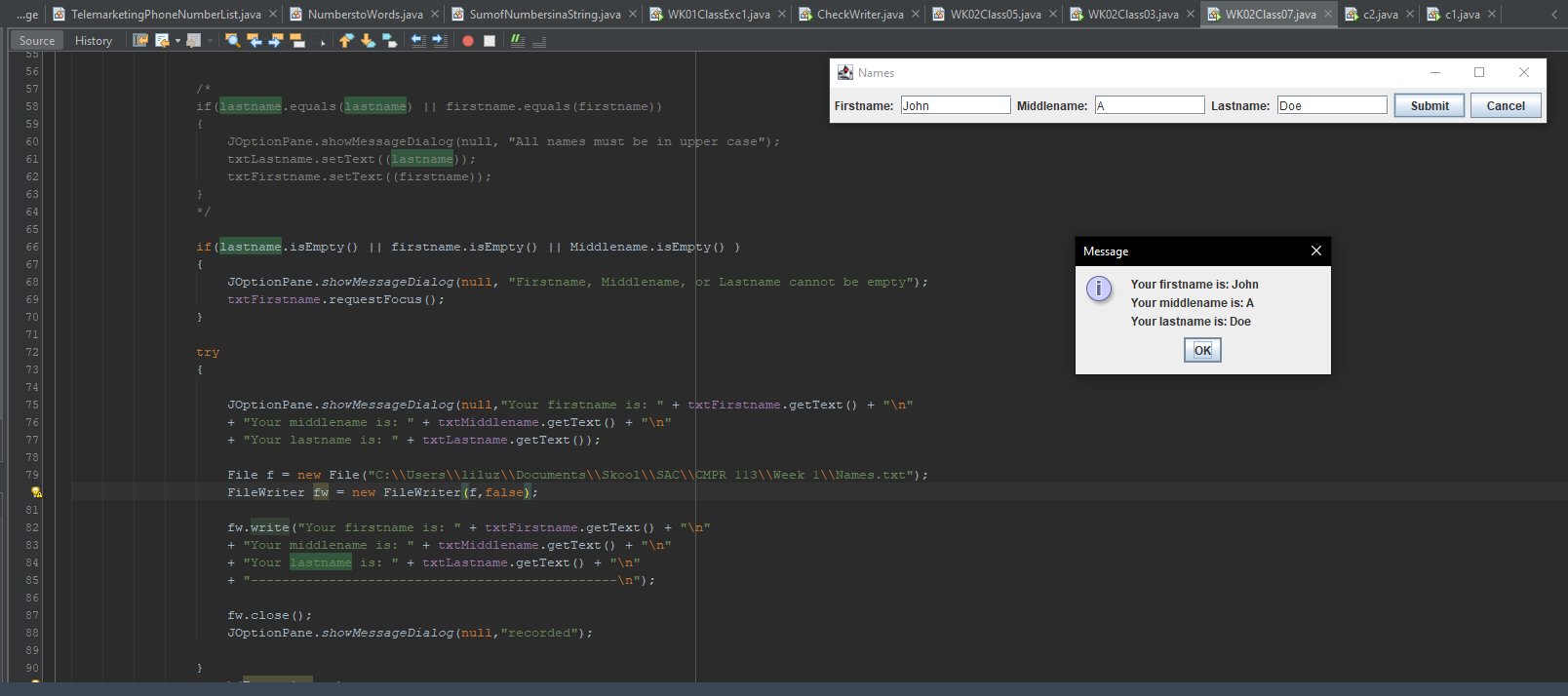
Reading from a file



**Project #7**

Create a class and name it **WK02Class07** and create a GUI application that will capture the last, middle and first name and output it to a file and then read the file

**#7 Print screen the running application with code below here**



**Submit this document to Module 1 Class Exercise #1**