1. Create a method ***ShortHnaded*** that receives a String and returns the String converted into shorthand. The simplified shorthand form of a string is defined as follows:
    a. Replace these four words: "and" with "&", "to" with "2", "you" with "U", and "for" with "4".
    b. Remove all vowels ('a', 'e', 'i', 'o', 'u', whether lowercase or uppercase), do not remove ‘u’ and ‘I’ if they did not occurred in a word.

```
public static String shortHanded(String str) {

        str = str.replaceAll("and", "&").replaceAll("to",
"2").replaceAll("you", "u").replaceAll("for", "4");
        str = str.replaceAll("[aeioAEOU]", "");
        String[] s = str.split(" ");
        str = "";
        for(int i = 0; i<s.length; i++) {
                if(s[i].length()>1) {
                        str += s[i].replaceAll("[uI]", "")+ " ";
                }else {
                        str += s[i] + " ";
                }
        }
        return str;

}
```

2. Create a Java method called numberOfSentences: a method that accepts a string and return number of sentences on it. A sentence can be determined if it ends with one of those punctuation (. , ? !).

```
public static int numberOfSentences(String str) {
        String[] arr = str.split("[.,?!]");
        return arr.length;

}
```

3. Create a Java method called numberOfWords: a method that accepts a string and return number of words, a word consists of more than 3 characters.

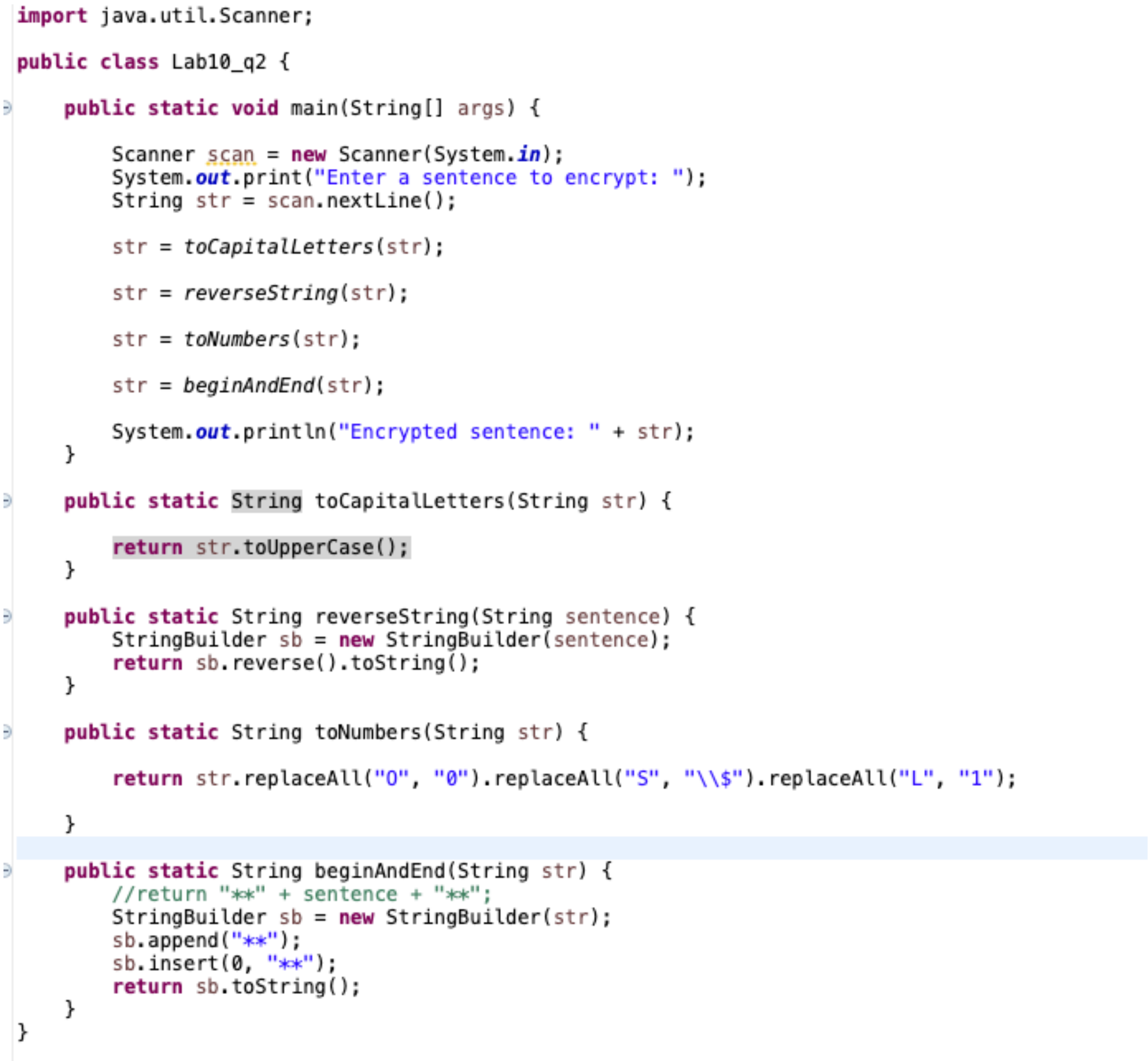
```
public static int numberOfWords(String str) {
        String[] arr = str.split(" ");
        int count = 0;
        for(int i = 0; i<arr.length; i++) {
                if(arr[i].length() > 3) {
                        count++;
                }
        }
        return count;

}
```

4. Design a java program that we can use to encrypt a sentence by using the following methods respectively:

   a. Convert al letters to capital.
   b. reverseString: reverse the sentence
   c. toNumbers: a method that converts the following letters to numbers: “O” to zero (0), “S” to $, “L” to 1
   d. Finally, you should display the encrypted sentence. beginAndEnd: that add ** at the beginning and at the end of the sentence.

Hint: use stringBuilder.



```java
import java.util.Scanner;

public class Lab10_q2 {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        System.out.print("Enter a sentence to encrypt: ");
        String str = scan.nextLine();

        str = toCapitalLetters(str);

        str = reverseString(str);

        str = toNumbers(str);

        str = beginAndEnd(str);

        System.out.println("Encrypted sentence: " + str);
    }

    public static String toCapitalLetters(String str) {

        return str.toUpperCase();
    }

    public static String reverseString(String sentence) {
        StringBuilder sb = new StringBuilder(sentence);
        return sb.reverse().toString();
    }

    public static String toNumbers(String str) {

        return str.replaceAll("O", "0").replaceAll("S", "\\$").replaceAll("L", "1");

    }

    public static String beginAndEnd(String str) {
        //return "**" + sentence + "**";
        StringBuilder sb = new StringBuilder(str);
        sb.append("**");
        sb.insert(0, "**");
        return sb.toString();
    }
}
```

Lab11-Q5

Consider you have the following Rectangle class:

Rectangle
- height: int - width: int
+ Rectangle(height : int, width: int) + getArea(): int + setWidth(width : int): void + setHeight(height : int): void + getWidth(): int + getHeight(): int

Write a java program to create an array of rectangle objects and read the information of the objects from the following file:

```
Rectangle_1 10 20
Rectangle_2 13 25
Rectangle_3 11 19
Rectangle_4 10 20
Rectangle_5 7 22
```

The first number represents the height and the second represent the width of the rectangle.

- Write the size of rectangles into a file called results.txt