**Name :** _____ **Date :** _____

# String Worksheet 2

**DIRECTIONS :** Fill in each blank with the correct answer/output. Assume each statement happens in order and that one statement may affect the next statement.

```
char cOne = 'o';
char cTwo = 'l';
String sOne = "hello there";
String sTwo = "hallo there";
```

```
out.print( sOne.indexOf('h') );          // LINE 1
out.print( sOne.indexOf('7') );          // LINE 2
out.print( sTwo.indexOf('a') );          // LINE 3
out.print( sTwo.indexOf(cTwo) );         // LINE 4
out.print( sOne.indexOf("lo") );         // LINE 5
out.print( sTwo.indexOf("al") );         // LINE 6
out.print( sOne.charAt(3) );             // LINE 7
out.print( sOne.charAt(0) );             // LINE 8
out.print( sOne.substring(3,6) );        // LINE 9
out.print( sOne.substring(0,4) );        // LINE 10
out.print( sOne.equals(sTwo) );          // LINE 11
out.print( sOne.compareTo(sTwo) );       // LINE 12
out.print( sTwo.compareTo(sOne) );       // LINE 13
out.print( sTwo.compareTo("abc") );      // LINE 14
out.print( sTwo.replaceAll("e","#"));    // LINE 15
out.print( sTwo.replaceAll("#","*"));    // LINE 16
out.print( sTwo.length() );              // LINE 17
out.print( sOne.length() );              // LINE 18
out.print('7' - '0');                    // LINE 19
out.print('5' - 48 );                    // LINE 20
out.print(sOne.charAt(20) );             // LINE 21
```

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____
16. _____
17. _____
18. _____
19. _____
20. _____
21. _____

**Write a method called `max` that takes in two whole numbers and sends back the bigger of the two numbers.**

**Write a method called `letterGrade` that will take in a decimal number and return the single character letter grade for that average (A, B, C, and F's only)**

**Write a method called `monkeyTrouble`. This method will have two `boolean` parameters called `aSmile` and `bSmile` that store whether or not each monkey is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return `true` if we are in trouble.**

**Write a method called `isClose` to determine if two circles are close to each other. This method assumes that both circles have a radius of `50` pixels. You will send the coordinates of both circles to this method and it will return whether or not they are close.**