Introduction to Java
Unit 5. Notes: Strings

The String Class. The String class represents a sequence of characters. A character can be a _____. Strings are unusual in that you do NOT have to use the _____ keyword or a _____ to instantiate an object of this class. For example: _____

A series of characters enclosed in quotes is called a _____
Each character has a unique location, starting at zero. The location of a character in a String is called its _____. For example:

String name = "Abe jumps"; // A is at index _____, j is at index _____

The **concatenation operator** is the plus sign (+). It is used to concatenate (join) two strings together. For example:
String word = "red";
String sentence = _____

String x = "3 days?"; int n = x.length(); _____ String y = "flower"; String a = y.substring(4); _____ String b = y.substring(1); _____ String c = "earth".substring(2); _____ String d = y.substring(1,2); _____ String e = y.substring(1,3); _____ String f = y.substring(0, 4) _____ String g = "box".substring(1,3); _____ for (int i = y.length() - 1; i > 2; i--) System.out.print(y.substring(i, i+1));	The length method returns an int equals to the number of characters in the string. What does the first line of the first substring method look like? public _____ substring _____ What does the first line of the second substring method look like? public _____ substring _____

```
String s = "heater";
int b = s.length();          _______
int c = s.indexOf( "a", 0 );     _______
int d = s.indexOf( "heat", 0 );  _______
int e = s.indexOf( "eat", 1 );   _______
int f = s.indexOf( "e", 2 );     _______
int g = s.indexOf( "ice", 0 );   _______
int h = s.indexOf( "a", 4 );     _______
```

What does the first line of the indexOf method look like?

public _____ indexOf _____

What does the indexOf method return if it cannot find what it is looking for?

```
String s1 = "4 me?";
String a = s1.toUpperCase();  _____________
String s2 = "HEY!";
String b = s2.toLowerCase(); _____________


if ( b == "hey!" )
        System.out.println( "ONE" );
else
        System.out.println( "TWO" );


if ( b.equals( "hey!" ) )
        System.out.println( "THREE" );
else
        System.out.println( "FOUR" );
```

What is the return type of the toUpperCase method?

public _____ toUpperCase()

What is the return type of the toLowerCase method?

public _____ toLowerCase()

If you need to check if two strings are equal,

do NOT _____

_____

public boolean equals(Object obj)

The equals method is case-sensitive.
"A" is not equal to "a"

The Scanner Class. Scanner objects are used to read streams of text. The text could come from the keyboard or a file. The table below describes three methods of the Scanner class.

Return Type	Method Name and Parameters	Description
double	nextDouble()	Ignores all white space and returns the first double it finds. It does not delete the new line character from the input stream.
int	nextInt()	Ignores all white space and returns the first integer it finds.
String	nextLine()	Returns all characters including white space up to, but not including, the new line characters. Deletes the new line character.

Some subtleties of the Scanner class. The Scanner actually deals with "streams" of data. For example, consider the code snippet below.

```
Scanner get = new Scanner(System.in);
System.out.print( "Enter the x and y coordinates " );
int x = get.nextInt();
int y = get.nextInt();
```

If the user types "7 8 9" (one space between each number) and then hits the Enter key, the input stream consists of _____ characters. What happens?

The first call to nextInt() _____

The second call to nextInt() _____

_____

```
System.out.print( "Enter your age and name " );
int age = get.nextInt();
String name = get.nextLine();
```

If the user types "16 Moe" and then hits the Enter key, the input stream consists of _____ characters. What happens?

The call to nextInt() _____

The call to nextLine() _____

_____