Method Calling and Return Types

Breakdown of a static method

3 parts we care about currently:
- Return Type: A promise that the return value of this method will be of the given type
- Name: Symbolic of what the method is going to achieve.
- Parameters: Data passed into the method to be used by it.

```
public static int foo (int i, double d, char c, String s)
                ^   ^     ^^^^^^^^^^^^^^^^^^^^^^
                |   |              Parameters
                | Name
           Return Type
```

Return Types

These can be any object the user has access to. Primitives and user-created objects included.

If a method doesn't return anything it will have the keyword void in place of a type. This is seen in *public static <u>void</u> main(String [] args)* and methods you have already created in previous assignments.

Method calling

Calling dot and a method name on an <u>object type</u> will call a static method in that object. Calling dot and a method name on an <u>object name</u> will call a method in that object, though it *may* not be static.

Method calls are evaluated from <u>LEFT TO RIGHT and PARAMETER FIRST</u>

Ex 1:
String s = "CaT"
s = s.toUpperCase().toLowerCase()   //s == "cat"

Ex 2:
String s = "Cat";
char c = s.charAt(Math.max(1, 2))   *//evaluate parameter Math.max first.*
                                             //this evaluates max of 1 and 2 and
                                             //returns 2. s.charAt(2) is t.
                                             //c == 't';

Returning in a method

The keyword *return* is used to send a desired return-value once a method has finished. This return-value must be of the promised type.

IMPORTANT: when a return is called, the method is instantly ended.
 Ex:
```
public static int print5(){
    return 5;
    System.out.println("5");        //This is never reached as a return has been called
}
```

MATH!

Math.max and Math.min

Math.max and Math.min both take 2 integers and return an int value. max returns the greater of the 2 ints and min returns the lesser.

Ex:
```
int j = Math.max(1, 2);        // j == 2
int k = Math.min(1, 2);        // k == 1
```

1. String s = "CaT"

*a)   s.toUpperCase();

*b)   s.toLowerCase();

c)   s.toLowerCase().toUpperCase();

d)   s.toUpperCase().toLowerCase().toLowerCase();

2.
*a)   Math.max(1, 2);

*b)   Math.min(1, 2);

c)   Math.max(Math.max(1, 3), 2);

d)   Math.min(Math.max(3, 4), Math.min(Math.max(3, 4), 4))

3. int j = 1; int k = 2;

*a)   j = Math.max(j, k);

*b)   j = Math.min(j, k);

*c)   j = Math.max(j, k);
       j = Math.min(j, k);

4)   int j, k = 2;   int m = 4;   String s = "CatSArecOOl"
s.toUpperCase().toLowerCase().toLowerCase().replace('A',
'p').substring(k).charAt(Math.max(Math.min(Math.max(Math.min(m, j), k), m), m));