CSC110 Lab10: Methods, Parameters, Return values **40 Points**

Submit on Canvas – See Canvas for Due Date

Write ***ONE application*** named, **Lab10.java**, that includes ***main*** and the following methods:

1. Method, ***power***, that takes two integers and returns the value of the first integer raised to the power of the second. Assume that the first parameter is greater than zero and that the second parameter is never negative. This method ***must NOT use any method in the Math class***. Rather, the power should be calculated using repeated multiplications.
2. Method, ***sumDigits***, that takes one integer and returns an integer that is the sum of the digits contained in the formal parameter's value. For example, if the formal parameter's value is 4982, the returned value should be 23, i.e., the sum of 2 + 8 + 9 + 4. This method should not use any Strings and not use any Java API methods. Use integer division, the remainder operator, and addition. *[Do not use any String variables, nor any String methods.]*
3. Method, ***substring***, that has the following parameter list: (String s, int start, int stop). This method returns a String containing a substring of the s that begins with the character at index start and ends with the character just before index stop. Example 1: if this method is called with the arguments, "weekend", 0, 4, then the String, "week", is returned. Example 2: if the arguments are: "hello", 1, 3, then the String, "el", is returned. This method should return null if start is not between zero and the length of s minus 1, inclusive, OR if stop is not between zero and length of s, inclusive. *[**Do not use any String class methods** except for length, charAt, and concatenation(+).]*
4. Method, ***devowel***, that takes a String value and returns that same String value but without its vowels. Example, the input "abc" would cause "bc to be returned; the input "iou" would cause the String, "", to be returned. *[**Do not** use any String class methods except for length, charAt, and concatenation(+).]*

Your main method should be used to test each of the exercise methods. Here are ***examples*** of tests that *could* be run from your main method:

```
// testing power
int x = 1;
int y = 0;
System.out.printf("%d raised to the power of %d is %d.\n", x, y, power(x, y));
x = 5; y = 1;
System.out.printf("%d raised to the power of %d is %d.\n", x, y, power(x, y));
x = 7; y = 2;
System.out.printf("%d raised to the power of %d is %d.\n", x, y, power(x, y));
x = 2; y = 8;
System.out.printf("%d raised to the power of %d is %d.\n", x, y, power(x, y));

// testing sumDigits
x = 1;
System.out.printf("Sum of the digits in %d is %d.\n", x, sumDigits(x));
x = 12345;
System.out.printf("Sum of the digits in %d is %d.\n", x, sumDigits(x));
x = 123456789;
System.out.printf("Sum of the digits in %d is %d.\n", x, sumDigits(x));

// testing substring
String s = "testing substring";
System.out.printf("Substring of <%s> from index %d up to index %d: %s.\n",
                  s, 8, 11, substring(s, 8, 11));
System.out.printf("Substring of <%s> from index %d up to index %d: %s.\n",
                  s, 0, 17, substring(s, 0, 17));
System.out.printf("Substring of <%s> from index %d up to index %d: %s.\n",
                  s, -1, 4, substring(s, -1, 17));
System.out.printf("Substring of <%s> from index %d up to index %d: %s.\n",
                  s, 0, s.length(), substring(s, 0, s.length()));
// testing deVowel
s = "abcdefg";
System.out.printf("%s devoweled became: %s.\n", s, deVowel(s));
s = "";
System.out.printf("%s devoweled became: %s.\n", s, deVowel(s));
s = "aeiou";
System.out.printf("%s devoweled became: %s.\n", s, deVowel(s));
s = "bcdgf";
System.out.printf("%s devoweled became: %s.\n", s, deVowel(s));
```

You **do not need to get user input** to test these methods from main.
Instead just call them with different argument values and display the returned values to see if they are correct, as is done above in the *example* tests that you can include in your main method.