Your task is to write a method evenSum that computes the sum of all even numbers between two given numbers. For example

Complete the following file:

Numbers.java

```
 1  public class Numbers
 2  {
 3      /**
 4          Computes a sum of even integers
 5          @param a the lower bound (may be odd or even)
 6          @param b the upper bound (may be odd or even)
 7          @return the sum of even integers between a and b (inclusive).
 8      */
 9      public int evenSum(int a, int b)
10      {
11          a = a + 0;
12          int sum = 0;
13          for (; a <= b; a++){
14              if (a % 2 == 0){
15                  sum = sum + a;
16              }
17              else {
18                  sum = sum + 0;
19              }
20          }
21          return sum;
22      }
23  }
```

Submit

Calling method

Testing method evenSum

	Arguments	Actual	Expected
pass	3,8	18	18
pass	4,8	18	18
pass	3,11	28	28
pass	4,11	28	28
pass	-10,2	-28	-28

Score

5/5

2019-07-01T18:12:22Z

When people enter their telephone numbers into a form they use many different formats. Some examples are:

```
(444) 999-1234
4440001234
444.999.1234
```

In the PhoneNumbers class below, complete the cleanNumber method. It takes a string and extracts the numeric digits from the string. If there are exactly 10 numeric digits, the number is re

Hint: Use the Character.isDigit method to check whether a character is a digit.

Complete the following file:

PhoneNumbers.java

```java
1
2  /**
3   * Write a description of class PhoneNumbers here.
4   *
5   * @author (your name)
6   * @version (a version number or a date)
7   */
8  public class PhoneNumbers
9  {
10     /**
11        Cleans a phone number.
12        @param phoneNumber a phone number that should contain ten digits and possibly other characters
13        @return the phone number in the form (###) ###-#### or the string "Error" if phoneNumber
14        does not have ten digits
15     */
16     public String cleanNumber(String phoneNumber)
17     {
18        String str = "";
19        for (int i = 0; i < phoneNumber.length(); i++)
20        {
21           char ch = phoneNumber.charAt(i);
22           if (Character.isDigit(ch))
23           {
24              str = str + ch;
25           }
26
27        }
28        if (str.length()==10){
29           String sub1 = str.substring (0,3);
30           String sub2 = str.substring (3,6);
31           String sub3 = str.substring (6,10);
32           return "(" + sub1 + ") "+sub2 + "-" + sub3;
33
34        }
35        else {
36           return "Error";
37        }
38
39     }
40  }
41
42
43
```

[Submit]

Calling method

Testing method cleanNumber

Arguments	Actual	Expected
pass (444) 999-1234	(444) 999-1234	(444) 999-1234
pass 4440001234	(444) 999-1234	(444) 999-1234
pass 444.999.1234	(444) 999-1234	(444) 999-1234
pass 444.999.123	Error	Error
pass 444.999.12345	Error	Error

Score

5/5