General information

This assignment is to be done on your own. See the Academic Dishonesty policy in the syllabus, [http://www.cs.iastate.edu/~cs227/syllabus.html#ad](http://www.cs.iastate.edu/~cs227/syllabus.html#ad), for details.

You will not be able to submit your work unless you have completed the Academic Dishonesty policy acknowledgement on the Homework page on Canvas. Please do this right away.

If you need help, see your instructor or one of the TAs. Lots of help is also available through the Piazza discussions.

Note: This is a miniassignment and the grading is automated. If you do not submit it correctly, you will receive at most half credit.

Overview

This is a short set of practice problems involving writing loops. You will write eight methods for the class `mini1.KeepMeInTheLoop`. All of the methods are static, so your class will not have any instance variables (or any static variables, for that matter). There is a constructor, but it is declared `private` so the class cannot be instantiated.

For details and examples see the online javadoc.

You do not need arrays or ArrayLists for this assignment, though you will not be penalized for using them. None of the problems requires nested loops.
Advice

Before you write any code for a method, work through the problem with a pencil and paper on a few concrete examples. Make yourself write everything down; in particular, write down things that you need to remember from one step to the next (such as indices, or values from a previous step). Try to explain what you are doing in words. Write your algorithm in pseudocode.

Another key problem-solving strategy is to try solving part of the problem, or solving a related, simpler problem. For example, here are some ideas for getting started on each problem using this strategy:

- To start solving `findNth`, can you
  a. iterate over a string and just print the characters one at a time?
  b. iterate over a string and print the index each time you see the given character ch?
  c. return the index of the first occurrence of ch?

- To start solving `doubleConsonants`, can you
  a. iterate over a string and just print the characters one at a time?
  b. iterate over a string and append each character onto a new string? (Tip: see the section below on creating a string with a loop)
  c. iterate over a string and append each character onto a new string, only if it isn't already the last character of the new string?
  d. iterate over a string and append each character twice onto a new string?

- To start solving `findStoppingTime`, can you
  a. write a conditional to check whether a number is even or odd?
  b. write statements to perform the indicated steps just once and print the result? (e.g. starting with n= 6, you'd get 3)
  c. write a loop to perform the indicated steps three times? (e.g. starting with n=6, you'd get 3, 10 and 5)

- To start solving `howLong`, can you
  a. write statements to perform the balance and monthly cost updates for one month, and print out the new values? How about two months?
  b. write a loop to deduct the monthly cost from the balance each month and print out the balance, assuming the interest rate and cost of living increase are zero?

- To start solving `isIBeforeE`, can you
  a. iterate over a string and print "boo" each time you find an 'e'?
  b. iterate over a string and print "boo" each time you find an 'e' with an 'i' after it (remember, the 'e' might be at the end of the string)
To start solving **findSecondLargest**, can you

a. parse the string and just print the numbers one at a time? (Tip: use a Scanner!)

b. parse the string and find the maximum value?

To start solving **isPermutation**, can you

a. check whether two strings are the same length?

b. iterate over the characters of s and check whether each character occurs somewhere in t? (Tip: use the String method `indexOf`)

c. combine (a) and (b) to solve the problem for the case when when s and t have no duplicates?

d. "remove" a character from string t? (Tip: you can do this with the `substring` methods, or using `java.util.StringBuilder`)

To start solving **containsWithGaps**, can you

a. iterate over the characters of `source` and check whether each character occurs somewhere in `target`? (Tip: use the String method `indexOf`)

b. once you find a character `c` in `target`, when you look for the next character, can you avoid characters to the left of `c`?

Remember that you can create helper methods if you wish, as long as you declare them `private`.

**My code's not working!!**

Developing loops can be hard. If you are getting errors, a good idea is to take a simple concrete example, and trace execution of your code by hand (as illustrated in section 6.2 of the text) to see if the code is doing what you want it to do. You can also trace what's happening in the code by temporarily inserting `println` statements to check whether variables are getting updated in the way you expect. (Remember to remove the extra `println`'s when you're done!)

Overall, the best way to trace through code with the debugger, as we are practicing in Lab 6. Learn to use the debugger effectively, and it will be a lifelong friend.

Always remember: one of the wonderful things about programming is that within the world of your own code, you have absolute, godlike power. If the code isn't doing what you want it to do, you can decide what you really want, and make it so. **You are in complete control!**

(If you are not sure what you want the code to do, well, that's a different problem. Go back to the "Advice" section.)
What do you mean, a private constructor?

Just put this declaration at the top of your class:

```java
/**
 * Private constructor disables instantiation.
 */
private KeepMeInTheLoop() { }
```

How do I make a string with a loop, as needed for doubleConsonants?

Start with an empty string and concatenate additional characters in each iteration. For example, here is one way to create the reverse of a given string:

```java
public static String reverse(String s)
{
    String result = ""; // start with empty string
    for (int i = s.length() - 1; i >= 0; i = i - 1)
    {
        result += s.charAt(i); // add on characters one at a time
    }
    return result;
}
```

As an aside, experienced Java programmers would probably use a StringBuilder object, which works like a mutable type of string:

```java
private static String reverse(String s)
{
    StringBuilder sb = new StringBuilder();
    for (int i = s.length() - 1; i >= 0; i = i - 1)
    {
        sb.append(s.charAt(i));
    }
    return sb.toString();
}
```

Testing and the SpecChecker

A SpecChecker will posted shortly that will perform an assortment of functional tests. As long as you submit the assignment correctly, your score will be exactly the score reported by the specchecker.

However, when you are debugging, it is much more helpful if you have a simpler test case of your own that reproduces the error you are seeing.
Remember that to call a static method, you prefix it with the class name, not with an object reference. For example, here is simple test case for the findNth method:

```java
import mini1.KeepMeInTheLoop;

public class SimpleTest {
    public static void main(String[] args) {
        System.out.println(KeepMeInTheLoop.findNth("mississippi", 's', 3));
        System.out.println("Expected 5");
    }
}
```

You can save yourself from having to type "KeepMeInTheLoop" over and over again by using the Java feature import static, which allows you to invoke a static method without typing the class name:

```java
import static mini1.KeepMeInTheLoop.*;

public class SimpleTest {
    public static void main(String[] args) {
        System.out.println(findNth("mississippi", 's', 3));
        System.out.println("Expected 5");
    }
}
```

Since no test code is being turned in, you are welcome to post your tests on Piazza for others to use and comment on.

**Documentation and style**

Since this is a miniassignment, the grading is automated and in most cases we will not be reading your code. Therefore, there are no specific documentation and style requirements. However, writing a brief descriptive comment for each method will help you clarify what it is you are trying to do.

**If you have questions**

For questions, please see the Piazza Q & A pages and click on the folder miniassignment1. If you don’t find your question answered, then create a new post with your question. Try to state the question or topic clearly in the title of your post, and attach the tag miniassignment1. But remember, do not post any source code for the classes that are to be turned in. It is fine to post
source code for general Java examples that are not being turned in. (In the Piazza editor, use the button labeled “pre” to have Java code formatted the way you typed it.)

If you have a question that absolutely cannot be asked without showing part of your source code, make the post “private” so that only the instructors and TAs can see it. Be sure you have stated a specific question; vague requests of the form “read all my code and tell me what’s wrong with it” will generally be ignored.

Of course, the instructors and TAs are always available to help you. See the Office Hours section of the syllabus to find a time that is convenient for you. We do our best to answer every question carefully, short of actually writing your code for you, but it would be unfair for the staff to fully review your assignment in detail before it is turned in.

Any posts from the instructors on Piazza that are labeled “Official Clarification” are considered to be part of the spec, and you may lose points if you ignore them. Such posts will always be placed in the Announcements section of the course page in addition to the Q&A page. (We promise that no official clarifications will be posted within 24 hours of the due date.)

**What to turn in**

**Note:** You will need to complete the "Academic Dishonesty policy questionnaire," found on the Homework page on Blackboard, before the submission link will be visible to you.

Please submit, on Canvas, the zip file that is created by the SpecChecker. The file will be named SUBMIT_THIS_mini1.zip and it will be located in the directory you selected when you ran the SpecChecker. It should contain one directory, mini1, which in turn contains one file, KeepMeInTheLoop.java. Always LOOK in the zip file the file to check.

Submit the zip file to Canvas using the Miniassignment1 submission link and verify that your submission was successful. If you are not sure how to do this, see the document "Assignment Submission HOWTO" which can be found in the Piazza pinned messages under “Syllabus, office hours, useful links.”

*We strongly recommend that you just submit the zip file created by the specchecker. If you mess something up and we have to run your code manually, you will receive at most half the points.*

We strongly recommend that you submit the zip file as created by the specchecker. If necessary for some reason, you can create a zip file yourself. The zip file must contain the directory mini1, which in turn should contain the file KeepMeInTheLoop.java. You can accomplish this by zipping up the src directory of your project. The file must be a zip file, so be sure you are using the Windows or Mac zip utility, and not a third-party installation of WinRAR, 7-zip, or Winzip.