File Practice Problems

1) Write a method reads in from two files and writes them, alternating lines, to an output file. The three file names should be passed into the method as `Strings`, and then method should NOT thrown any exceptions (i.e. all exceptions from opening files should be handled within this method)

   For example, if the first input file is:
   ```
   This is the first input file.
   It contains three lines.
   This is the last line.
   ```

   and if the second input file is:
   ```
   This is the second input.
   It only contains two lines.
   ```

   Then this method should create the following file as output:
   ```
   This is the first input file.
   This is the second input.
   It contains three lines.
   It only contains two lines.
   This is the last line.
   ```

```
public static void alternateLines(String inFileName1, String
                                  inFileName2, String outFileName) {
     Scanner inFile1 = null;
     Scanner inFile2 = null;
     PrintWriter outFile = null;
     try {
          inFile1 = new Scanner(new File(inFileName1));
          inFile2 = new Scanner(new File(inFileName2));

          outFile = new PrintWriter(outFileName);
     }
     catch (FileNotFoundException e) {
          System.out.println("Error: " + e);
     }

     while(inFile1.hasNextLine() || inFile2.hasNextLine()){
          // both infiles still have lines left to read
          if (inFile1.hasNextLine() && inFile2.hasNextLine()) {
               outFile.println(inFile1.nextLine());
               outFile.println(inFile2.nextLine());
          }
```

```
            // only infile 1 has lines left to read
            else if (inFile1.hasNextLine()) {
                 outFile.println(inFile1.nextLine());
            }
            // only infile 2 has lines left to read
            else {
                 outFile.println(inFile2.nextLine());
            }
       }
       inFile1.close();
       inFile2.close();
       outFile.close();
  }
```

2) Write a method which reads in a file, and writes the first 5 lines of this input file to a different output file. The input and output file names should be passed into the method as Strings. If there are less than 5 lines in the input file, this method should write all of them to the output file. This method should NOT throw any exceptions.

For example if the input file is:

```
This is the input file.
There
is
one
word
per
line
here.
```

Then the output file would be:

```
This is the input file.
There
is
one
word
```

If the input file is:

```
hello!
```

then the output file would be:

```
hello!
```

```
public static void writeFirst5(String infileName,
                               String outfileName) {
```

```
        Scanner inFile = null;
        PrintWriter outFile = null;

        // open up the infile and outfile
        try {
             inFile = new Scanner(new File(infileName));
             outFile = new PrintWriter(outfileName);
        }
        catch (FileNotFoundException e) {
             System.out.println("Error: " + e);
        }

        // initialize a counter to 0
        int count = 0;
        // loop while the counter is less than 5
        // and there are still lines in the infile
        while(inFile.hasNextLine() && count <5){
             outFile.println(inFile.nextLine());
             // increase the counter
             count++;
        }
        inFile.close();
        outFile.close();
}
```

3. Write a method that reads in a file and writes each of the lines to a new output file, with the `String` "`line i is:`" before the line, where `i` is actually the line number of the input file. Both the input and output file names should be passed into the method as `String`s. This method should NOT throw any exceptions.

   For example, if the input file is:

   ```
   Once upon a time,
   there was a frog
   who wanted to be a
   computer programmer.
   ```

   Then the output file would be:

   ```
   line 1 is: Once upon a time,
   line 2 is: there was a frog
   line 3 is: who wanted to be a
   line 4 is: computer programmer.
   ```

```
public static void labelLines(String infileName,
                              String outfileName) {
     Scanner inFile = null;
     PrintWriter outFile = null;

     // open up the infile and outfile
     try {
          inFile = new Scanner(new File(infileName));
          outFile = new PrintWriter(outfileName);
     }
     catch (FileNotFoundException e) {
          System.out.println("Error: " + e);
     }

     // initialize a counter to 0
     int count = 1;
     // loop while there are still lines in the infile
     while(inFile.hasNextLine()){
          outFile.println("line " + count + " is: " +
                                              inFile.nextLine());
          // increase the counter
          count++;
     }
     inFile.close();
     outFile.close();
}
```

4) Write a method that reads in two files and writes only the lines that are the same in the two files to an output file. Note that you are only comparing the lines at the same position in the files. Both the input and output file names should be passed into the method as `Strings`. This method should NOT throw any exceptions.

For example, if the first input file is:

```
The first line is the same.
Second line 1
cow
Fourth line
The fifth line is also the same.
```

and the second input file is:

```
The first line is the same.
Second line 2
```

```
Third line
cow
The fifth line is also the same.
```

Then the output file would be:

```
The first line is the same.
The fifth line is also the same.
```

```
public static void sameLines(String inFileName1,
                    String inFileName2, String outFileName) {
     Scanner inFile1 = null;
     Scanner inFile2 = null;
     PrintWriter outFile = null;
     try {
          inFile1 = new Scanner(new File(inFileName1));
          inFile2 = new Scanner(new File(inFileName2));

          outFile = new PrintWriter(outFileName);
     }
     catch (FileNotFoundException e) {
          System.out.println("Error: " + e);
     }

     // We only have to loop while both files have lines,
     // since if only one in file has lines remaining,
     // they won't match the (non-existent) lines of the
     // other file.
     while(inFile1.hasNextLine() && inFile2.hasNextLine()){
          String line1 = inFile1.nextLine();
          String line2 = inFile2.nextLine();
          if (line1.equals(line2)) {
               outFile.println(line1);
          }
     }
     inFile1.close();
     inFile2.close();
     outFile.close();
}
```