Refactoring in Python

Instructions

The assignment requires that you use functions, types and classes from the Python standard library.

Solution format: implement the required script in a file `refactor.py`, adequately commented.

Script Specifications

Write a Python script `refactor.py` that takes two arguments: a URL and a path. The URL must point to a Java source file on a *remote* host; whereas the second argument must be the path of a *local* file containing a Java snippet. This snippet can be assumed to be the declaration of a single `static` method that takes an `Object` as argument and returns a `String`. Note that you should **not** make any assumption on the method name, i.e. methods in different snipped files can have different names.

Your script must do the following:

1. download the *remote* file: you can assume that the file contains the definition of a single class;
2. copy the snippet inside the class defined in the *remote* file;
3. perform a refactoring of all the invocations of `System.out.println` with an argument, by wrapping such argument with a call to the method copied in the previous point. For example, suppose that the snippet method is called `capitalize` and consider the statement

   ```
   System.out.println("the result of math.sqrt(2) is " + Math.sqrt(2.0));
   ```

   your refactoring should result in the statement

   ```
   System.out.println(capitalize("the result of math.sqrt(2) is " + Math.sqrt(2.0
   ```

4. compile the refactored file; if the compilation fails report an error message to the user, and redirect the compiler messages to the file `COMP_ERR.txt` before terminating (this file must be create only if compilation errors occur);
5. if the compilation succeeds, run the resulting Java class and save the output of the execution in the file `OUTPUT.txt`, and possible error messages in the file `ERROR.txt`, then terminate.

You can test your script using the snippet files `snip1.java` and `snip2.java` (available on the course webpage). However, the code of your script should be parametric and should not make any assumption on the names of the Java file and of the method in the snippet file.

Finally, your script **must** handle anomalous situations, e.g. when the script is invoked with less arguments than expected; or when the URL is invalid; or the needed files do not exist.

Hint: Here are some Java files, you can use to test your script:

- ClassWithTest.java
- Main.java
- Client.java
- insertion_Sort.java