Python Ecosystem Assignment 1
Packages and Frameworks

1. Create a basic testing framework in python that supports the following features
    a. The runner (runner.py)
        i. Write functions to search through the root directory (specified in the command line arguments) containing multiple .py files to find all test functions, run them and to print the stats.
        ii. Assumption: All the test functions start with the substring “test_a1_”.
        iii. Hint: Use the os, inspect, importlib.machinery modules

    b. Test Structure and Assertion Library
    Implement the test_a1_has_sample() test function in the test1.py file, which does the following
        i. You read pre-existing text files and assert that all files contain the substring “sample” . Use context managers to implement the with statement for file handling. The assert errors and the file handling related exceptions must be handled by the context manager. It is assumed that all the text files will exist in the “textfiles” directory which is present in the same directory as test1.py.
        ii. The assert function to be used in the test (assert_has_sample()) must be implemented in the assertions.py file
        iii. Design a custom error class “SampleNotFoundError” for the above test by extending the runtime class and throw this error in the assert function if “sample” is not found in a text file.
        iv. The test succeeds if all files in the textfiles directory contain the substring sample, else it fails.

    c. Implement a simple decorator function_call_log(), which logs each function call in runner.py, i.e. decorate all your functions in the runner.py using this decorator.

2. Package your framework as a distribution package and upload it to TestPyPI
    a. Note that the code in the external_code directory won’t be a part of the package
    b. The package must be named hw_package_<your_names>
    c. You must create both a wheel and .tar.gz file.
    d. Include a license, the author name, email, the long and short description in the metadata, that will be reflected on the TestPyPI platform.

In the end, running the **python3 -m hw_package_<yournames> external_code** command should run the test correctly.

3. Deliverables
    a. Source code
    b. Link to your package on test.pypi.org

Note : Please refer to the boilerplate code provided.