

Software Metrics

Prof. Yordanka Budinova

Assignment 1

Due date: 23rd November 2021

**Prepared by**

|  |  |
| --- | --- |
| Name: | Ario Anindito |
| Student ID: | 2016082 |
| Programme: | Bachelor of Science  (Software Engineering) |

**Software Metrics Assignment 1 – Report One**

The purpose of this report is to outline the calculation results for the following metrics: LOC, PLOC, and Comments. I used Radon as a tool to accomplish.

Radon is a Python tool which computes various code metrics. Supported metrics are:

* Raw metrics: SLOC, comment lines, blank lines, &c.
* Cyclomatic Complexity (i.e. McCabe’s Complexity)
* Halstead metrics (all of them)
* The Maintainability Index (a Visual Studio metric)

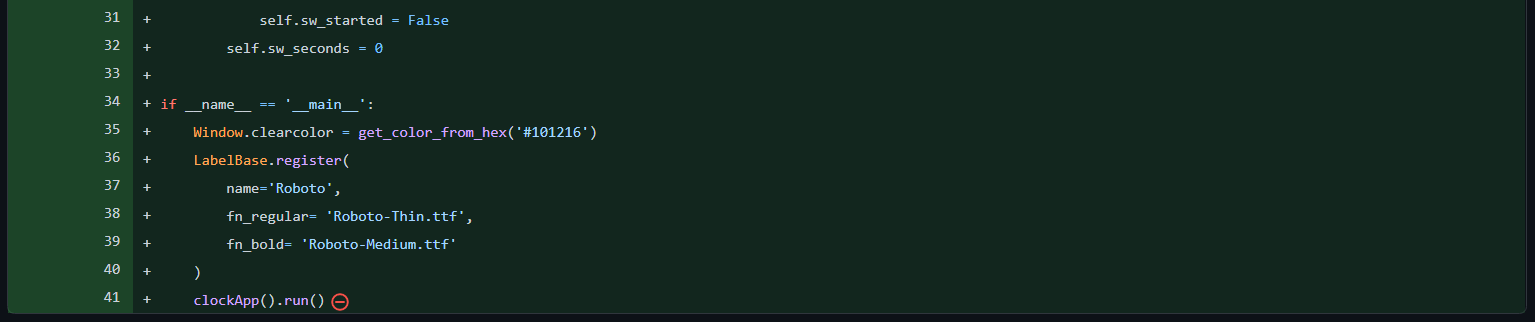
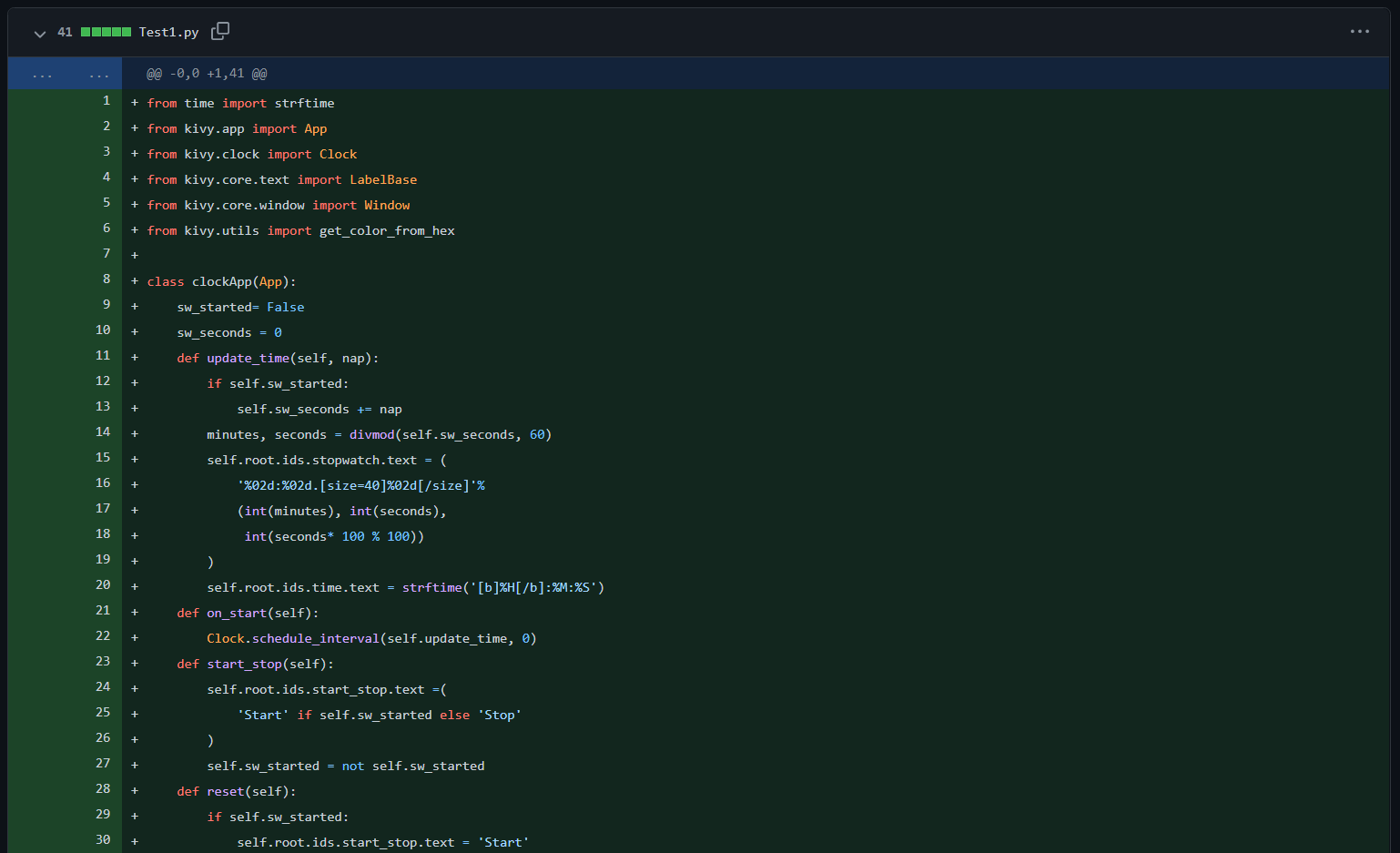
Raw command in Radon analyzes the given Python modules in order to compute raw metrics. These include:

* LOC: the total number of lines of code
* LLOC: the number of logical lines of code
* SLOC: the number of source lines of code - not necessarily corresponding to the LLOC [Wikipedia]
* comments: the number of Python comment lines (i.e. only single-line comments #)
* multi: the number of lines representing multi-line strings
* blank: the number of blank lines (or whitespace-only ones)

Finally, I used pylint to get the review/score in term of bug and quality check for the code, then made an improvement afterwards.

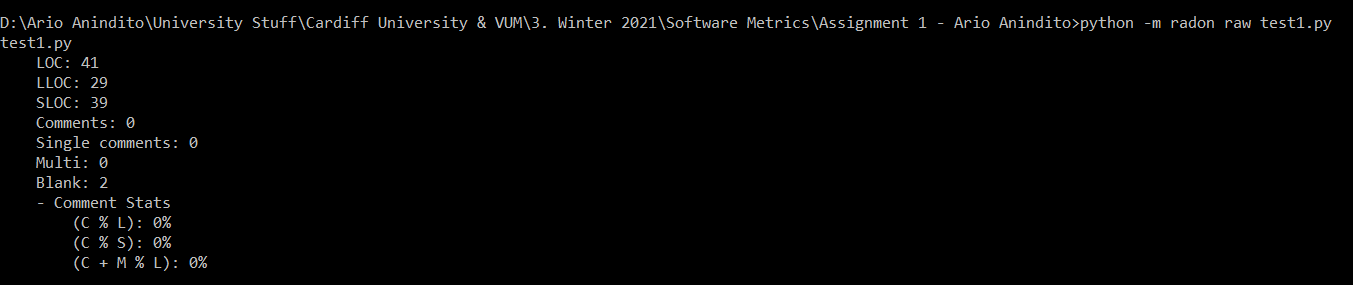
I put all the file project inside my repository <https://github.com/arioanindito>

Code before improvement



The code is all about a simple clock app with python using the kivy module in python.

The result for metrics and pylint results are followed (Respectively):

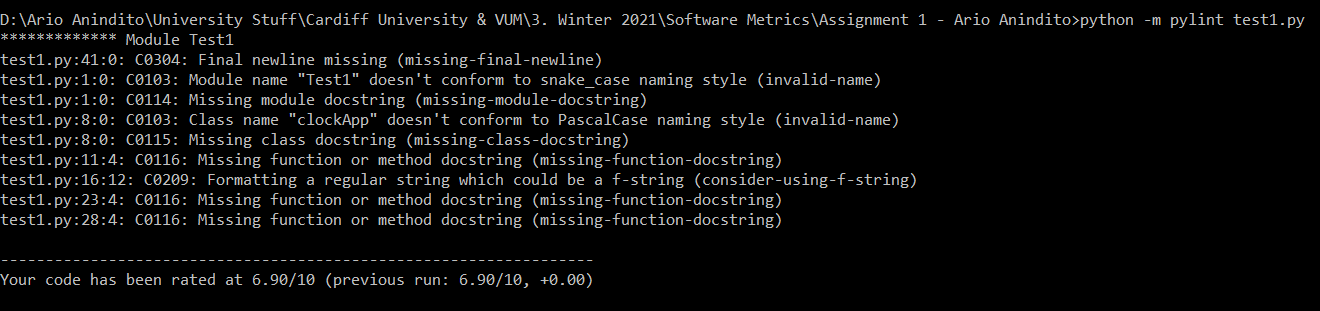


The metrics result (before improvement)

LOC: 41

LLOC: 29

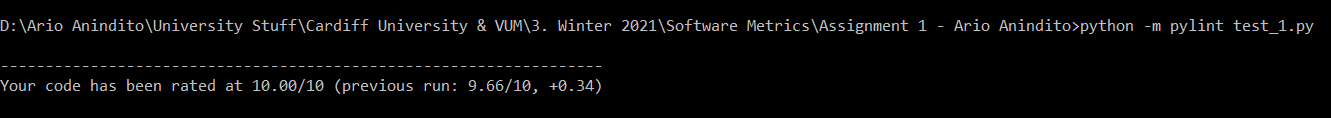
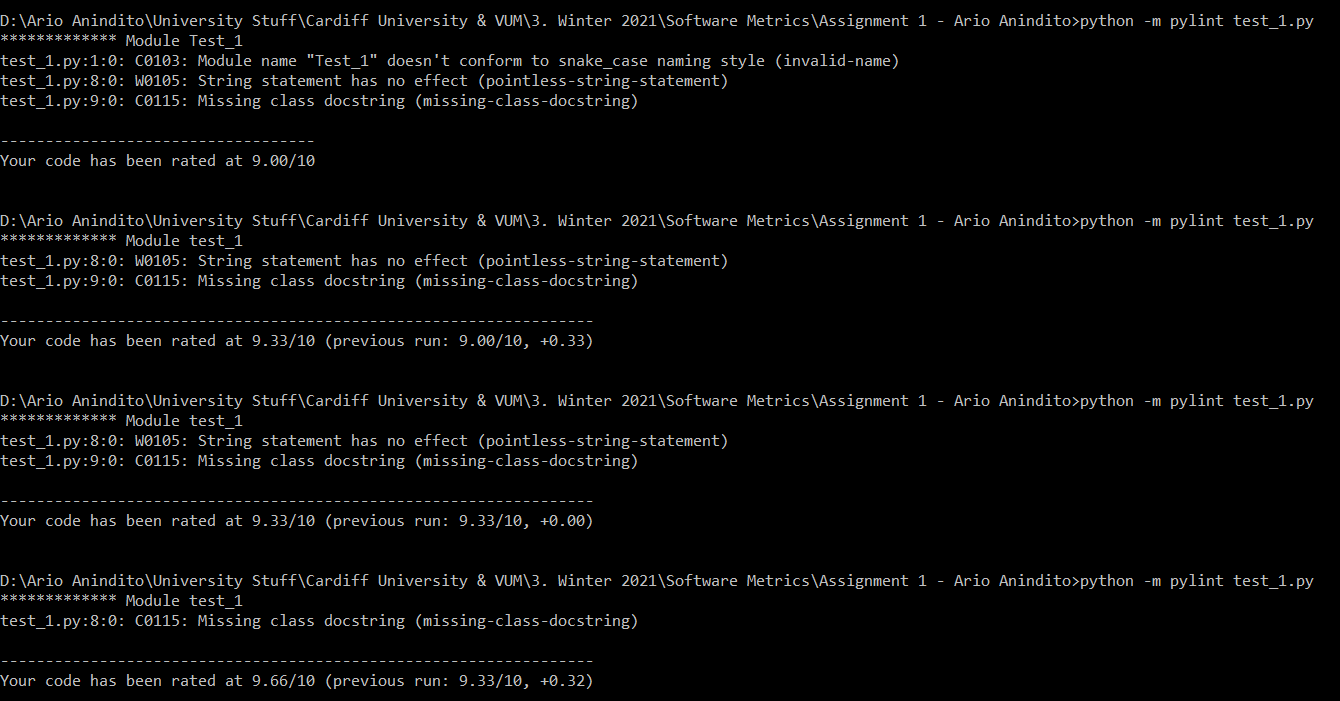
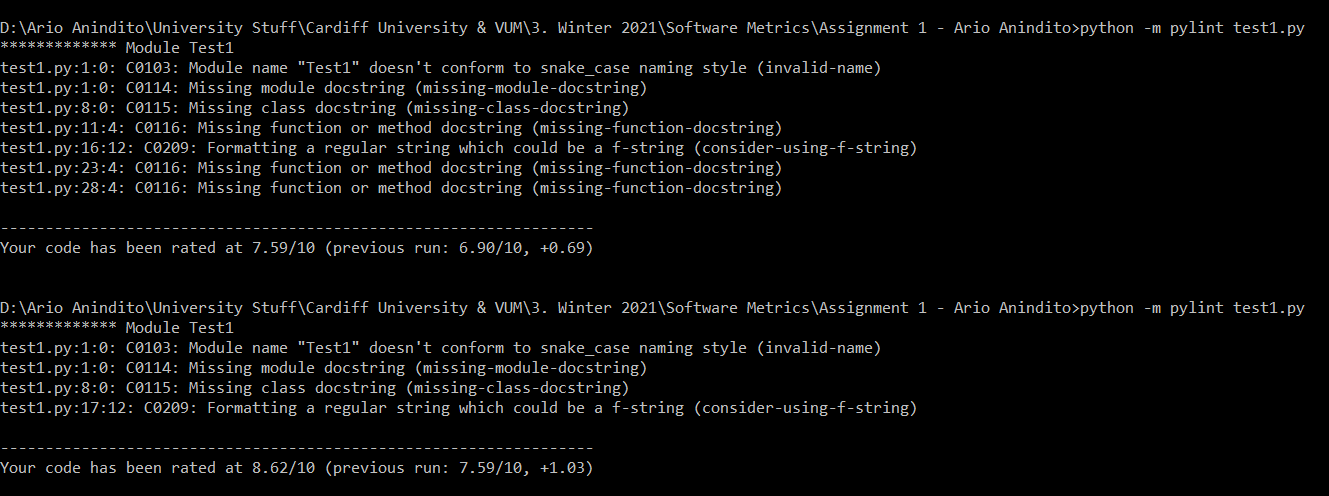
Comments: 0



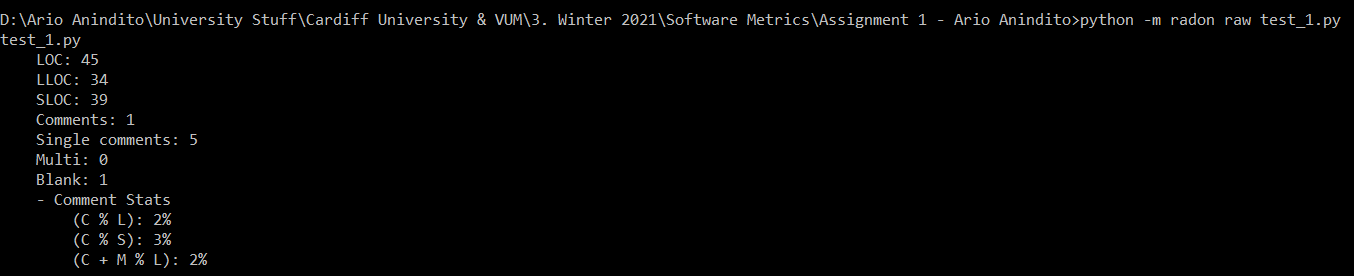
The pylint result (before improvement)

Code has been rated at (6.90/10.00)

I made an improvement a couple times, and these are the following result:

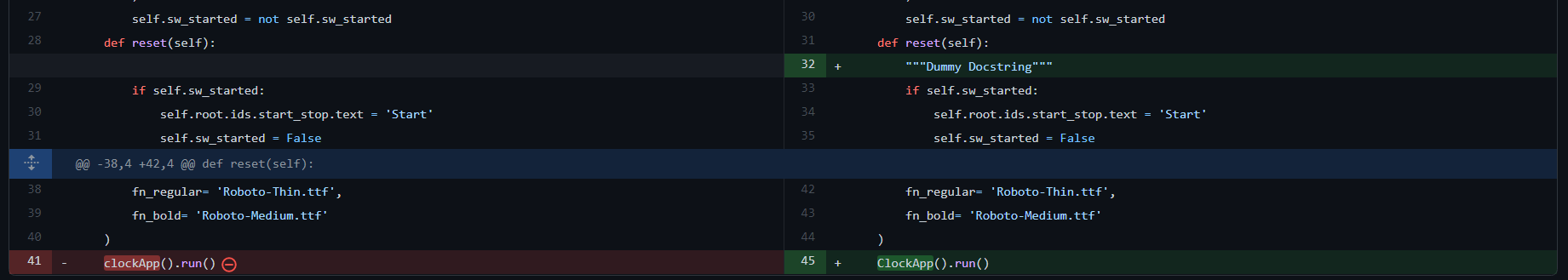
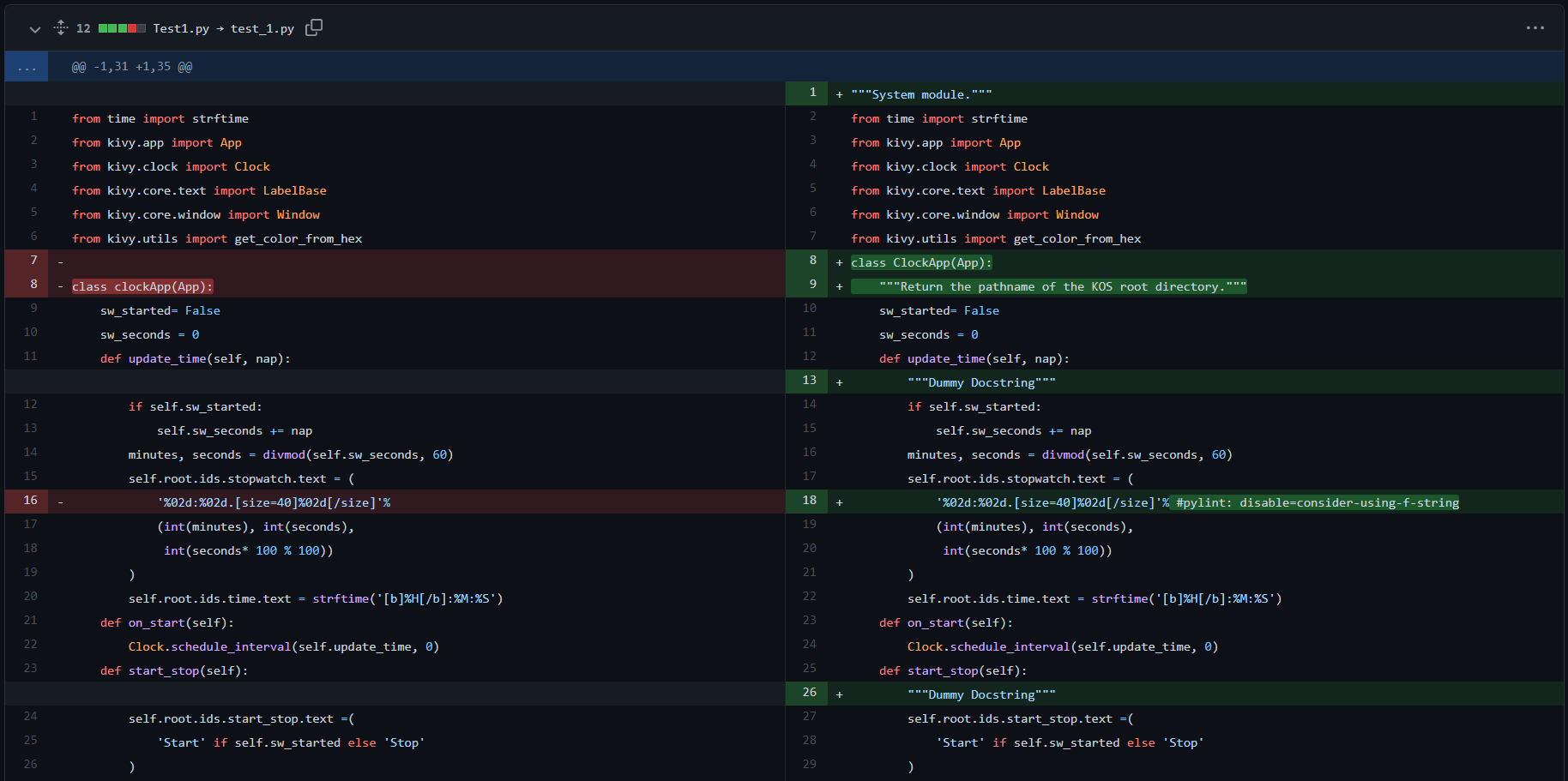


At the end, the result for pylint become 10.00/10.00, with the radon result are followed:



Moreover, if we compare side by side the code before and after improvement we got the following result:

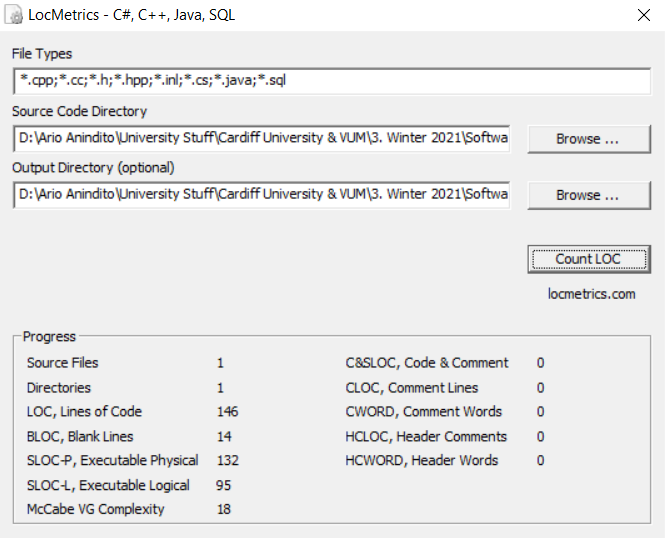
<https://github.com/arioanindito/swm2021/commit/4177faa45a3ec1bcd1e468c79250ab80b792c8dd>



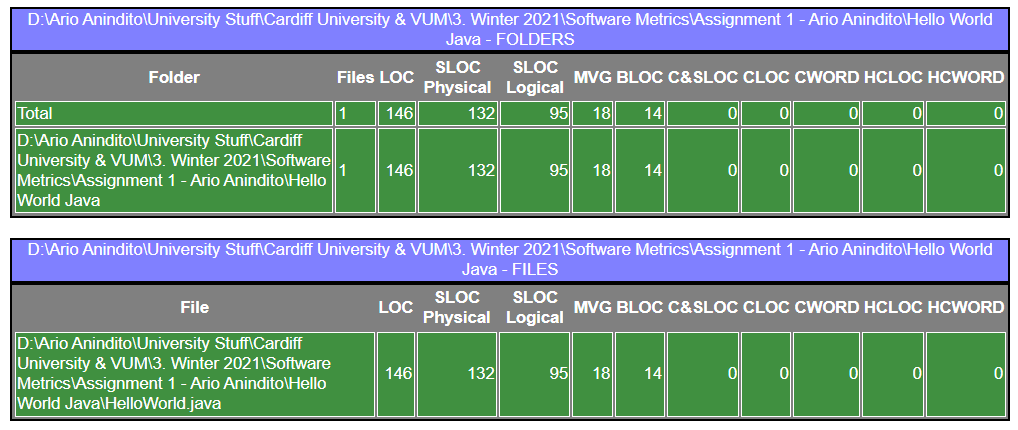
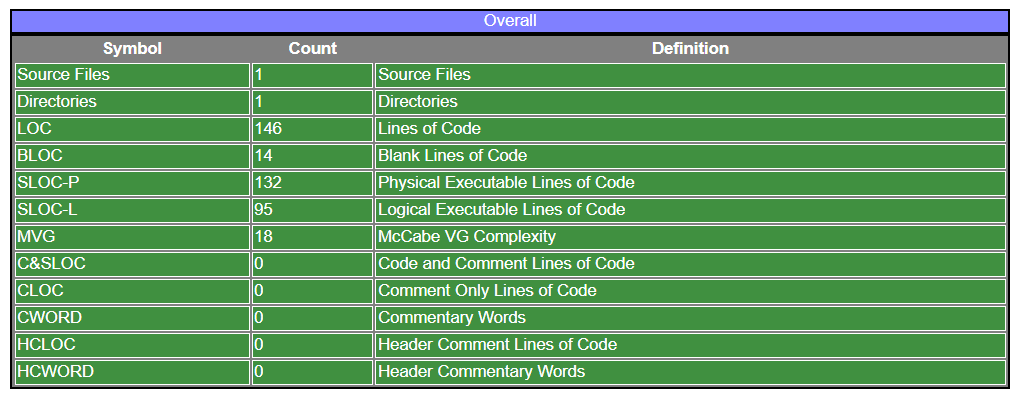
**Software Metrics Assignment 1 – Report Two**

The purpose of this report is to outline the calculation results for the following metrics: LOC, PLOC, and Comments. I used Loc Metrics as a tool to accomplish.

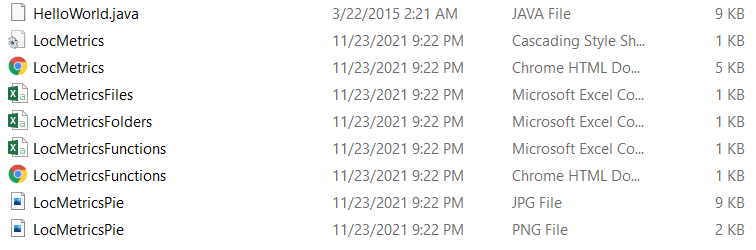
The overall user interface of Loc Metrics are followed:



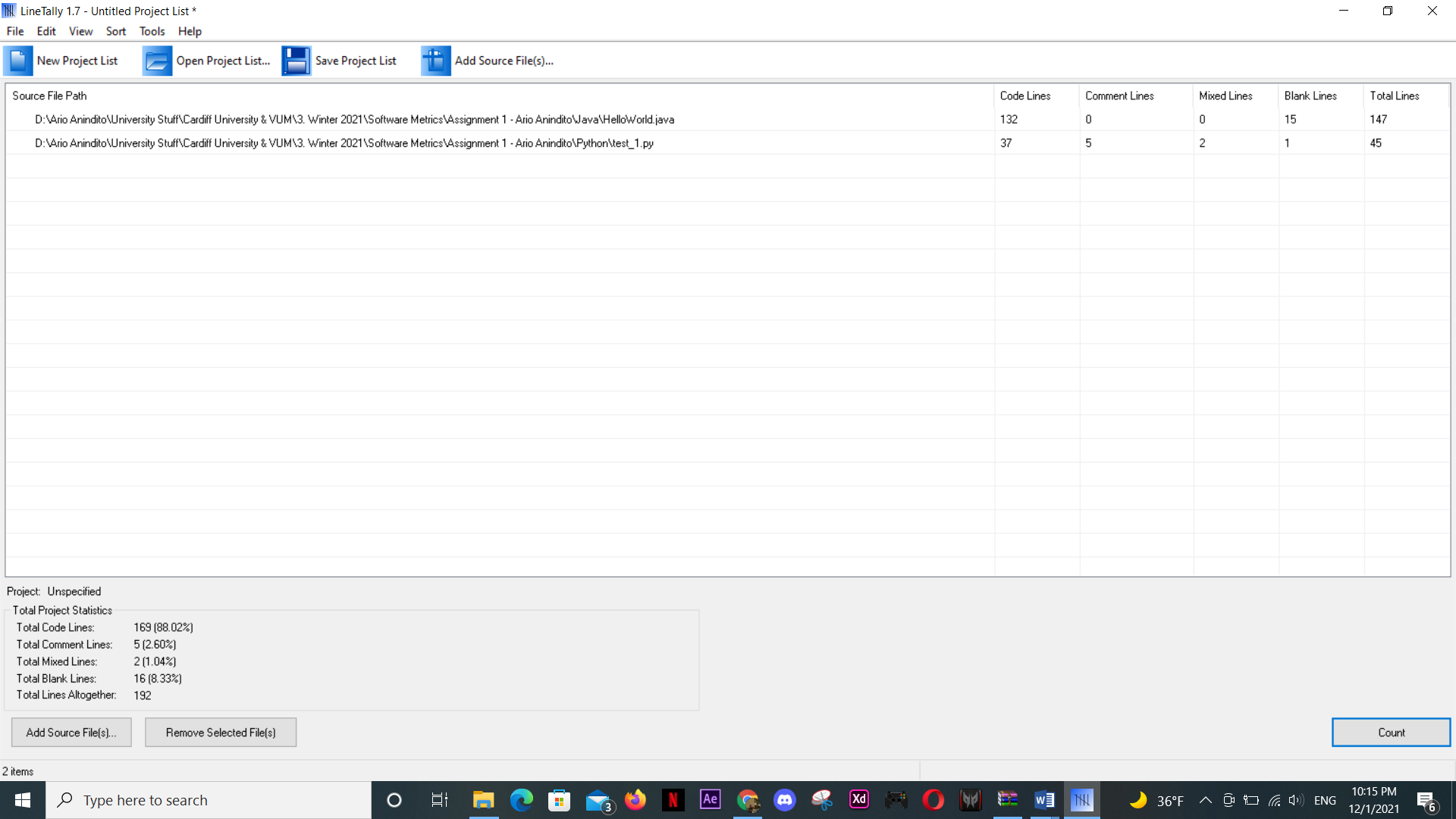
After choosing the directory that I put the source code in, I just need to click “Count LOC” button to begin measurement. Loc Metrics created the results as a report files to the directory as shown below.



The program also generate analyzed reports using pie charts and tables as shown below.



Furthermore, in order to count lines for both of the project (Python and Java) I used Line Tally for a convenient. I add both project as a source files and count it. It gave me the results as shown below:



Conclusion

In this project I used four different tools to accomplish. Radon, Pylint, Loc Metrics, and Line Tally. All the purpose of the tools are for measurement.

Radon is a Python tool which computes various code metrics. Pylint is a source-code, bug and quality checker for the Python programming language. Loc Metrics and Line Tally are the software tool to measure source lines of code.

I put all the files and results in my github repository,

<https://github.com/arioanindito/swm2021/tree/master/Assignment%201%20-%20Ario%20Anindito>

I made a commit for my custom python code for improvement,

<https://github.com/arioanindito/swm2021/commit/4177faa45a3ec1bcd1e468c79250ab80b792c8dd>