Re-engineering a legacy system

This assessment is worth 10% of the final grade of PR301 and is due on Thursday 24 September at the start of the class on that day. You must hand in all required analysis documents and all code.

THIS ASSIGNMENT INVOLVES:

 The study and documentation of the logic and data structures of an existing program, and then re-engineering the program in an alternative language (Python) using current "best practice" programming techniques.
 This is an INDIVIDUAL Assignment. Each student must work on their own DIFFERENT program.

ACTIVITIES TO PERFORM:

Study the supplied program listing and create:
1. A **summary document** describing the overall aim and requirement of the program and the expected processing outcomes.
5% ❍❍❍❍❍
2. A **data glossary** defining all variables in the original program. For each variable including the name, the data type, range of possible values (if applicable), a list of all line numbers on which the variable is used, and an appropriate fuller name for the variable. (*Note: where a variable is 'recycled' in the original program for several different purposes it should be treated as if it was several DIFFERNT variables and each should be documented separately.)*
10% ❍❍❍❍❍ ❍❍❍❍❍
3. Draw a diagram which shows the **data structures** used in the program.
5% ❍❍❍❍❍
4. Indentify all the **major processing routines** in a listing of the original program. For each routine provide a meaningful routine name, identify starting and ending line numbers, and identify every place in the original program from where the routine is called.
5% ❍❍❍❍❍
5. Document the **overall program logic** by drawing a Structure Diagram showing the relationship between all the major processing routines.
10% ❍❍❍❍❍ ❍❍❍❍❍
6. For each processing routine document **the intent of the logic** using pseudocode.
10% ❍❍❍❍❍ ❍❍❍❍❍
7. Write Python doctests or unittests which will act as **acceptable tests** that verify the output of the re-engineered program and its major routines is correct.
15% ❍❍❍❍❍ ❍❍❍❍❍ ❍❍❍❍❍
8. **Re-engineer** the program in Python. *Use methods but not classes. Use of global variables is acceptable.*
30% ❍❍❍❍❍ ❍❍❍❍❍ ❍❍❍❍❍ ❍❍❍❍❍ ❍❍❍❍❍ ❍❍❍❍❍
9. Document the re-engineered program with PEP8 compliant **docstrings**. *(Note: DOCSTRINGs used for documentation are required as well as DOCTESTS for testing. They are different!)*
5% ❍❍❍❍❍
10. Ensure that the naming conventions and layout of the final Python program is **PEP8** compliant.
5% ❍❍❍❍❍