

CSIT121 Assignment 3

Visualized Analysis for Green Star Projects

Due Date: Week 13

Marks: 10

1. Objective

1. The objective of Assignment 3 is to apply the objective-oriented design and programming knowledge learned from Week 1 to Week 12 lectures. In this assignment, students will design a Python program, utilizing UML class diagrams, to assist the Green Star in providing a visualized analysis of registered and certified projects (see Fig. 1).



Fig. 1: An example of the visualized analysis of Green Star projects

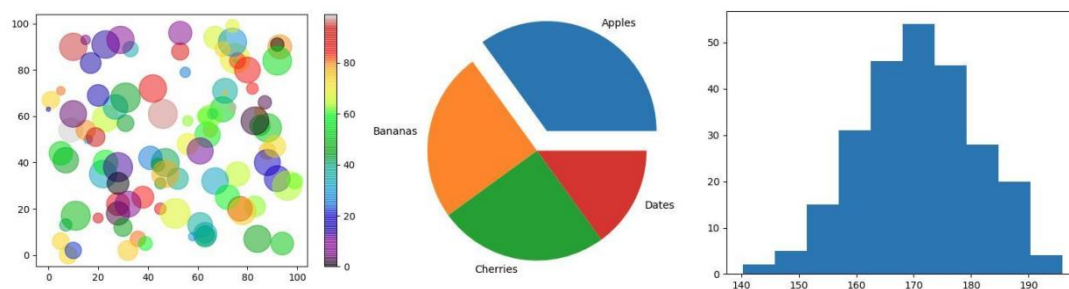


Fig. 2: Examples of visualization styles in Python

The proposed Python program should facilitate object serialization and deserialization processes (using modules such as pickle or JSON), allow for visualization (using matplotlib) of the analysis results using different styles (see Fig. 2), and enable the import and export of visual analysis results for printing, reuse, and documentation purposes. Students are required to apply at least one Python class design pattern (e.g., decorator pattern and singleton pattern) introduced in the lecture in this assignment to gain a better understanding of object-oriented programming code principles in practical scenarios.

2. Project Description

To provide a comprehensive visual analysis of Green Star projects, students are asked to design a Python program that fulfils the following requirements:

2.1 Design

1. Create a UML class diagram to represent the object-oriented design of the Python program, incorporating different class design patterns.
2. Develop suitable classes to capture information related to Green Star projects, companies, registered and certified dates, rating tools, etc., utilizing appropriate class relationships and OOD, such as inheritance and polymorphism.
3. Implement object serialization and deserialization processes to enable the persistence of object data.

2.2 Program Implementation

The general steps for this project are as follows:

1. Collect data on the project's rating tools, locations, issued dates, etc., on the Green Star Project Directory (<https://www.gbca.org.au/project-directory.asp>).
2. Collect user's inputs to perform visual analysis, utilizing different visualization styles, such as bar charts, line charts, pie charts, etc., to present the analysis results as informative and visually appealing.
3. Enable the object serialization and deserialization (using JSON or pickle) to save and load the user's data.
4. Enable the import and export functionalities (using File I/O) to allow users to save and retrieve visual analysis results for printing, reuse, and documentation purposes.

2.3 Testing and Analysis

1. Design and implement test cases using the Python unittest module to verify the correctness of the implemented functionalities (at least one test per class).
2. Analyze the test results and evaluate the effectiveness of the applied class design patterns in achieving code reuse and modularity. Students should indicate which design pattern is applied and how it is used.

3. Tasks

This assignment consists of three tasks:

1. Students must examine the Green Star projects on the Green Star project directory (<https://www.gbca.org.au/project-directory.asp>) using OOA, and complete the class design using class design patterns and UML class diagrams.
2. Students must implement Python code based on their design and evaluate their programs with multiple projects. The Python program should be capable of collecting user inputs, generating multiple types of charts using matplotlib based on user inputs, handling unexpected user inputs, and finishing the object serialization and deserialization using pickle or JSON. Unittest is required to test the correctness of the Python program.
3. Students need to describe their object-oriented design, including class design, the design patterns used, attributes and methods, exception handling mechanism, unittest, and coverage analysis, in a PDF report. Also, the report is expected to include screenshots of the running results to verify the correctness of visualization functionality.

4. Submission

Students must submit the following components to Moodle Assignment 3 as a zipped file.

4.1 Report Submission

1. Provide a PDF file (named visualization.pdf) describing the program's functionality, class designs, and the class design patterns used.
2. Include a UML class diagram representing the object-oriented design of the Python program, highlighting the applied class design patterns.
3. The report must provide clear instructions on running the program, and it must include screenshots of the running results, to verify the correctness of the functionalities.
4. The report must explain the unittest and code coverage in detail.

4.2 Source Code Submission

1. Submit the complete Python program (named A3.py), implementing the visualized analysis system, adhering to the visualization requirements.
2. Submit test cases developed using the Python unittest module, along with an analysis of the test results and insights into the effectiveness of the applied class design patterns.

5. Appendix

- A. The Australian Sustainable Development Goals (SDGs) website: <https://sdgs.org.au>
- B. Green Star website: <https://new.gbca.org.au>
- C. Green Star Project Directory: <https://www.gbca.org.au/project-directory>