

# CSIT121 Assignment 3

## Visualized Analysis for Pokémon Pokédex

Marks: 10

Late Submission Penalty: 5% per day, 0 marks after 7 days. The email submission will NOT be marked.

### 1. Objective

The objective of Assignment 3 is to practice object-oriented design and programming skills acquired during Weeks 1 to 12. In this assignment, you are required to design and implement a Python program to assist the Pokémon Pokédex in visualizing summary reports based on user requests (see Fig. 1 for examples).

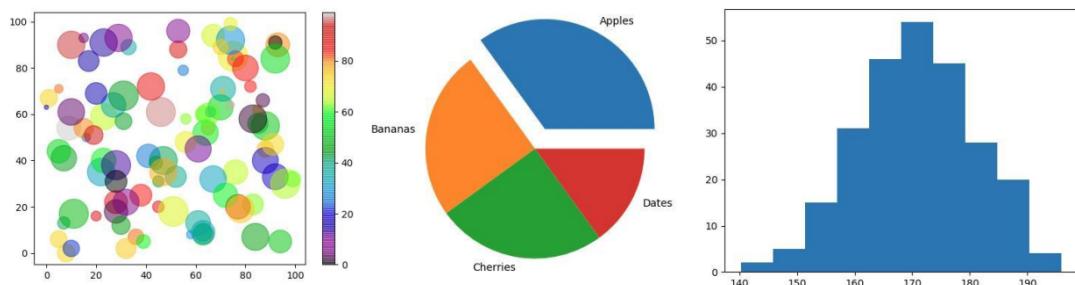


Fig. 1: Examples of visualization styles in Python

The proposed Python program should facilitate object serialization and deserialization processes (using **JSON**), allow visualization of analysis results with different styles (using **Matplotlib**), and enable the export of visualized analysis results for documentation. You are required to apply at least one Python design pattern (e.g., decorator pattern, singleton pattern) to enhance your understanding of object-oriented programming principles in practical scenarios.

### 2. Project Description

The Python program should gather user requirements to generate a summary report containing both textual information and figures, with at least three different visualization styles (e.g., bar chart, pie chart, or line chart). The program must robustly handle user inputs using Python's exception handling mechanisms (try...except). Additionally, it should support object serialization/deserialization using JSON for saving and loading objects.

You are expected to complete the following tasks based on your solution from Assignment 2 or the provided reference solution:

#### 1. Update of Assignment 2 Solution:

- Update your UML class diagram and code to include object serialization/deserialization and data visualization.

#### 2. File I/O Operations:

- If the JSON file does not exist, use file I/O operations to import project details from ‘Pokémon.txt’ and create project instances (same as Assignment 2).
- Users should be able to view, search for, create, and modify Pokémons (same as Assignment 1). Changes should be saved to both Pokémon.txt and Pokémon.JSON when the user exits the program.

### 3. Object Serialization/Deserialization:

- If the JSON file exists, the program should be able to load the JSON file and use it to re-create Pokémons objects.
- The program should be able save all Pokémons objects and export them to a JSON file.
- JSON format example:

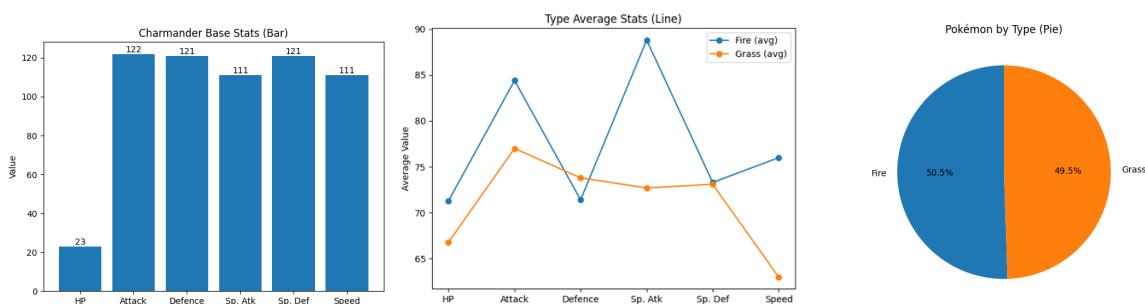
```
{
  "name": "Charmander",
  "hp": 23,
  "attack": 122,
  "defence": 121,
  "special_attack": 111,
  "special_defense": 121,
  "speed": 111
}
```

### 4. Regular expression validation for Pokémons attributes:

- Add a new option that allows users to modify Pokémons advanced information, i.e., national number, height, and weight.
- Implement input validation using Python’s **regular expressions** for Pokémons height (e.g., **1.2 m**, a float with exactly one digit after the decimal point and the unit must be **m**), Pokémons weight (e.g., **13.4 kg**, a float with exactly one digit after the decimal point and the unit must be **kg**), and national number (e.g., **No. 0034**, a ‘No.’ prefix followed by a space and exactly four digits).
- The input must match case-sensitive regular expressions exactly.

### 5. Matplotlib Visualization:

- The program should be able to generate figures to show the statistical analysis. For example, the program can generate a bar chart for a Charmander to show its attributes (e.g., HP, attack, defense, etc.). Also, a line chart is well-suited for comparing trends across the same ordered attributes between two groups (Fire vs. Grass averages). You could choose any visualization styles and any data to generate figures.
- At least **three** different visualization styles, such as bar chart, line chart, pie chart, etc., should be used to present the statistical analysis results.



#### **6. Implement a Menu System:**

- Implement a menu using a ‘while’ loop to present reasonable options to the user, such as creating a new project, searching for projects, and exporting a report.
- Users should be able to exit the program by typing “exit” or “X.”

#### **7. Testing:**

- Use the unit test module to test your object serialization and deserialization functions and generate a code coverage report for your test results.

### **3. Steps to Complete the Project**

Please follow the object-oriented software development methodology to complete your project.

#### **Step 1. Object-Oriented Analysis (OOA):**

Analyse the Pokémon data available on the Pokémon website to identify key classes representing Pokémon details such as name, number, type, and HP, for the purpose of generating text-based reports. Identify at least three classes and describe your analysis process, including how you identified classes, attributes, and methods, in your report.

#### **Step 2. Object-Oriented Design (OOD):**

Create a UML class diagram to illustrate your class design for this project. Your class design should contain at least three classes and should also use the inheritance and polymorphism. You can use the Python build-in exception classes or define your own exception class to handle the user’s invalid inputs. Your class diagram should show class attributes, methods, associations and generalisations and use appropriate prefixes to indicate private/public attributes and methods, as well as static attributes and methods (if applicable). Explain your UML class diagram in your report.

#### **Step 3. Object-Oriented Programming (OOP):**

Implement your Python program based on your UML class diagram and test your program using the unit test and coverage modules.

### **4. Submission Guidelines**

You are required to submit a PDF report and all source code, including the project module, the unit test module and the JSON file, to the Moodle Assignment 3 section. Email submissions will NOT be marked.

#### **4.1 PDF Report Submission (3 marks)**

1. Submit a PDF report named “Assignment3\_report.pdf” that explains your process for identifying classes and attributes in this project. The report must explicitly name the chosen design pattern, explain the intent, and show where it is applied in the code (class/method references).
2. Include a UML class diagram created using professional tools like UMLet. Hand-drawn diagrams are not

acceptable. The diagram should detail class attributes, methods, and relationships. You should clearly explain which design pattern is used in your report.

3. Include screenshots of your program in operation, showcasing key functions: (1) collecting user requests, (2) handling invalid requests, (3) collecting new Pokémon values (i.e., national number, height and weight) from the user and handling invalid inputs, (4) displaying imported Pokémon from both text file and JSON file, (5) updating the `pokemon.txt` file after creating new Pokémon or modifying existing Pokémon, (6) exporting a JSON file, (7) creating figures as the statistical analysis based on the search results, and (8) the unit test and coverage analysis results.

#### 4.2 Source Code Submission (7 marks)

Implement the source code using IDLE or your preferred IDE. Submit the project module as “`A3.py`”, the unit test module as “`test_A3.py`”, and the JSON file as “`pokemon.json`” via Moodle. Your source code should contain sufficient comments to explain class attributes and methods.

## 5. Appendix

Pokémon Pokédex Website: <https://pokemondb.net/pokedex/all>