Practical Activity 4: Developing a Python Utility Library with Git

Date issued: 19 March 2024

Objective:

To apply Python programming skills in developing a utility library while reinforcing version control and collaboration using Git and GitHub. In this activity students should be able to demonstrate a deep understanding of:

- Decision making and repetition.
- Application of data structures such as lists and data frames.
- Sub-setting data by rows and columns with a use of relational operators.
- Create functions that returns a value and functions that do not return a value.
- Create functions with parameters and arguments.
- Create lambda functions.
- Create comprehensions: lists, dictionaries, sets and tuples.
- Apply the basic concepts of data analysis in a program.

Requirements:

- Completion of Practical Activity 2 and Practical Activity 3.
- Python 3.x and Git installed on your computer.
- Basic understanding of Python programming concepts as listed and Git commands (clone, branch, add, commit, push, pull, merge).

Overview of the Library Management System:

The Library Management System will manage books and members, allowing books to be added, borrowed, and returned. It will support the following features:

- Book Management: Add, update, and delete book entries.
- Member Management: Add, update, and delete member entries.

Setup Instructions:

- **Repository Setup**. Clone this repository (https://github.com/xpiyose/library-management-system) into your local machine. Write this command:

  ```
  git clone https://github.com/xpiyose/library-management-system
  ```

- **Branch Creation**: Create a new branch for feature development, use any name. Hint: You will have to create a directory to store your entire project.

Task Distribution

- **Task 1**: Data Structures Implementation (lists and data frames for managing books and members).
- **Task 2**: Comprehensions: lists, dictionaries and sets
- **Task 3**: Data analysis using python

Task 1: Data Structures Implementation (lists and data frames for managing books and members).

Initialize two data structures to keep track of **books** and **members** both represented as lists. The system features two functions (You must create these functions): **add_book** and **add_member**. The **add_book** function takes three parameters (**book_id, title**, **author**, **status**) and appends a new book dictionary to the books list. The **add_member** function, on the other hand, requires two parameters (**member_id**, **name)** and appends a new member dictionary to the members list. Each member dictionary includes an empty list for borrowed_books to track the IDs of books each member has borrowed.

Sequence hint: initialize two variables as lists, then create two functions as per the above requires. An example of the appending part of the question is as follows:

```
books.append({
        "book_id": book_id,
        "title": title,
        "author": author
    })
```

You must use the same procedure for all appends!

Task 1 A:

Suppose you use the system to add a book titled "*Python Programming"* written by *Jacob Zuma* with a *book_id* of *2024001,* and a *member* named *Anelisa Maleka* with a *member_id* of *1*. How would these additions reflect in the books and members lists, and what would the output look like if you printed both lists immediately after these additions?

Hint: call the functions and write a print statement for them

Task 1 B:

Rewrite the entire task 1 and task 1B without using parameters and arguments in your functions.

Task 1 C:

Rewrite the entire task 1B without using functions

Task 1 D:

Rewrite the entire Task 1 C using Data frames instead of lists.

Task 2: Comprehensions (lists and sets).

The libraries is using the codes 14, 15, 16, 17, 18, 19, 20 to all programming related books:

- Create a normal and comprehensive list that will display the codes.
- Create a normal and comprehensive list that will add the codes together for auditing purpose.
- Create a normal and comprehensive list that will display only codes that are tracked by odd numbers.
- Create a set to display the list of codes.

Task 3: Data analysis using python (lists and data frames for managing books and members).

You are provided with insurance dataset on blackboard. Please logon on blackboard and download the dataset. Write a python code to:

- Read the dataset.
- Inspects its column by displaying the first 10 records.
- Display records for make and usage for sets_num that are more than 40.
- Plot a basic graph showing effective_yr on y axes and carrying capacity on x-axes

Submission Guidelines

- Add xpiyose@gmail.com as a collaborator for your project.
- Push the final version of your code to the shared repository.