To-Do List Application with Python, OOP, and Data Visualization

Using Python, Object-Oriented Programming (OOP), and Matplotlib for Visualization

Presented by : ASHISH KUMAR(246102002)

CONTENT:

Introduction

- **Objective**:
- To create a task management application (To-Do List) that allows users to:
    - Add, delete, and mark tasks as complete/incomplete.
    - Visualize task completion status with a pie chart.
- **Tools and Libraries Used**:
- Python, Matplotlib, Object-Oriented Programming (OOP) principles

Project Overview

- **Features**:
- Add and delete tasks.
- Mark tasks as complete/incomplete.
- Display all tasks with their completion status.
- Generate a pie chart of completed vs. incomplete tasks.
- **Programming Concepts**:
- OOP (Classes, Methods)
- Data visualization with Matplotlib

Class Structure

- **Classes**
- **Task Class**: Represents an individual task.
- **To Do List Class**: Manages a list of Task objects and provides functionalities like adding,
  deleting, and marking tasks complete.
- **Explanation**:
- Each task has properties: title and is completed.
- To Do List class manages all Task objects and tracks their status.

Code: Task Class

•**Code Snippet**:

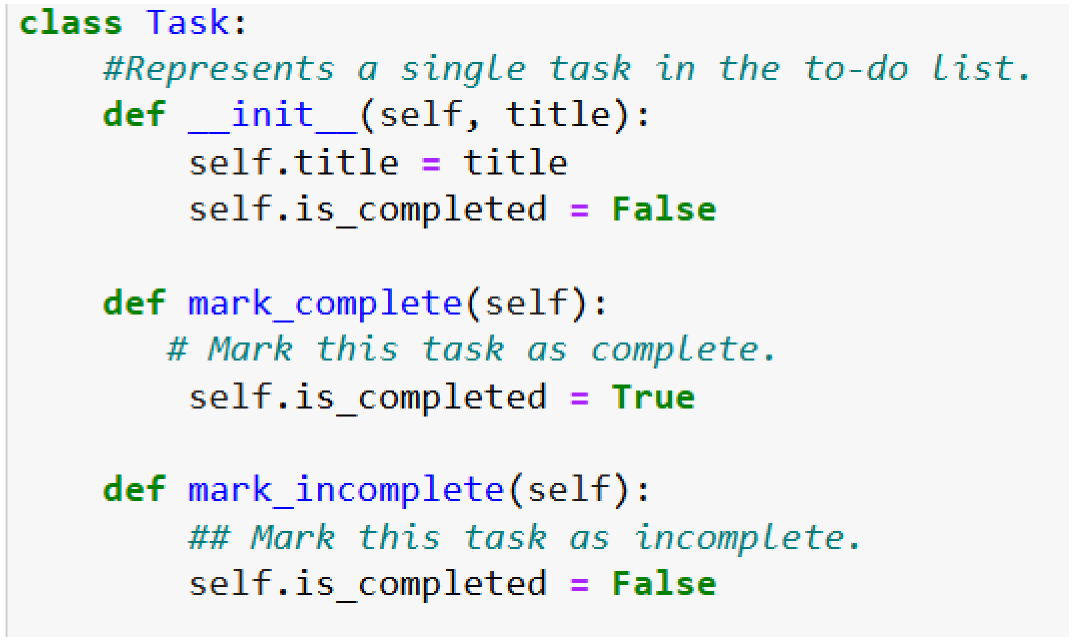
•**Explanation**:

•Initializes each task with a title.

•Marks tasks as complete or incomplete.

•**Purpose**:

•Simplifies task creation and state management.



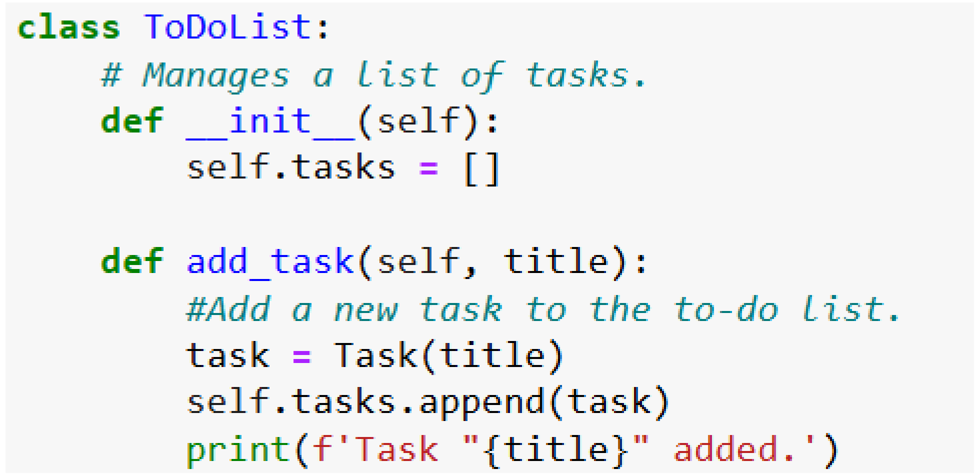
```
class Task:
    #Represents a single task in the to-do list.
    def __init__(self, title):
        self.title = title
        self.is_completed = False

    def mark_complete(self):
      # Mark this task as complete.
        self.is_completed = True

    def mark_incomplete(self):
        ## Mark this task as incomplete.
        self.is_completed = False
```

To Do List Class

Code Snippet:



```
class ToDoList:
    # Manages a list of tasks.
    def __init__(self):
        self.tasks = []

    def add_task(self, title):
        #Add a new task to the to-do list.
        task = Task(title)
        self.tasks.append(task)
        print(f'Task "{title}" added.')
```

Explanation:

- Initializes a list to hold tasks.
- Has methods for adding, deleting, marking complete, and displaying tasks.

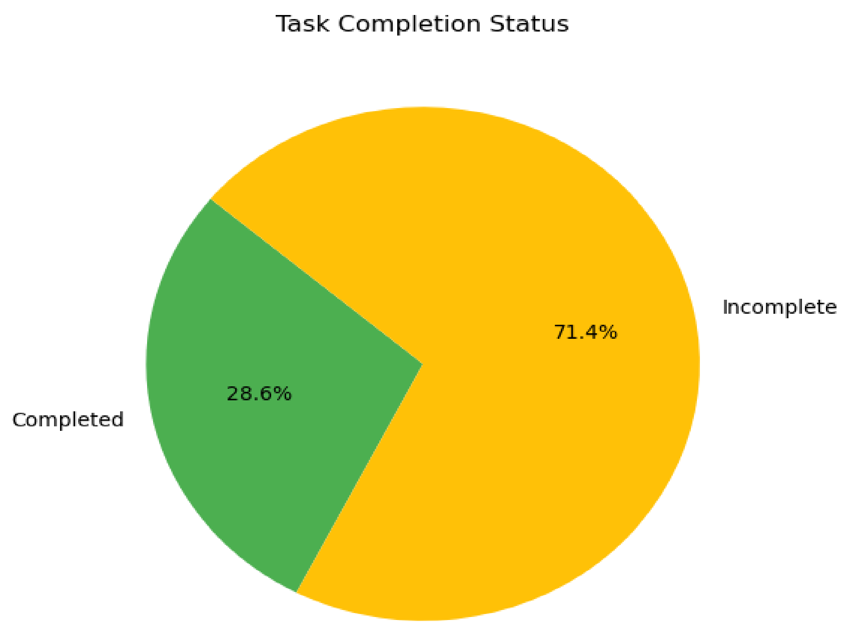
User Interface in Code

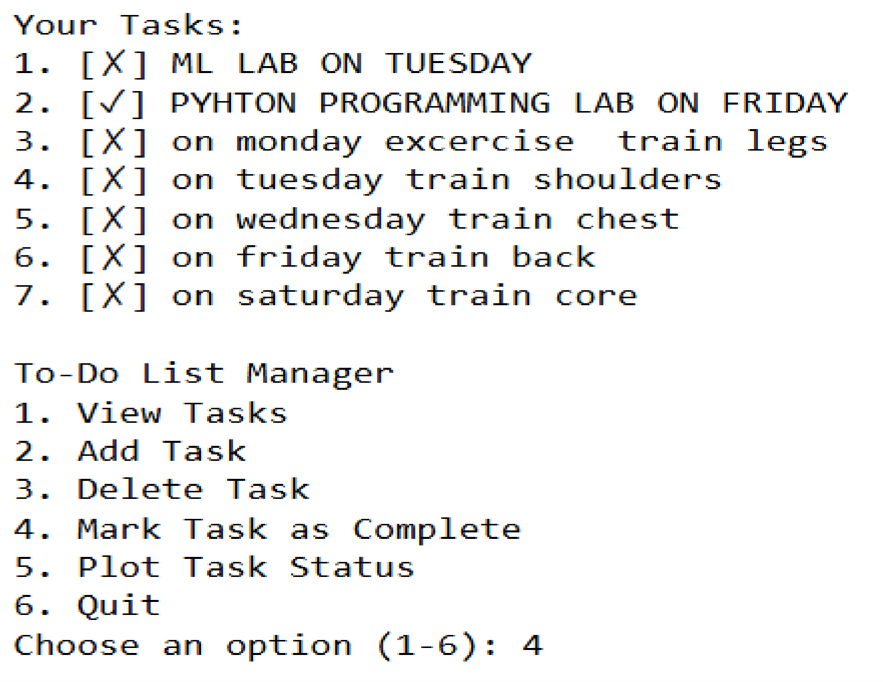
- **Methods**:
- display_tasks(): Displays each task with a check or cross.
- add_task(), delete_task(), mark_task_complete(): Modifies tasks.
- **Explanation**:
- User interacts with these methods in the console.

Data Visualization

Pie Chart with Matplotlib:

Generating a pie chart to show completed vs. incomplete tasks.(Plot showing the percentage of tasks completed and incompleted.)





```
Your Tasks:
1. [X] ML LAB ON TUESDAY
2. [✓] PYHTON PROGRAMMING LAB ON FRIDAY
3. [X] on monday excercise  train legs
4. [X] on tuesday train shoulders
5. [X] on wednesday train chest
6. [X] on friday train back
7. [X] on saturday train core

To-Do List Manager
1. View Tasks
2. Add Task
3. Delete Task
4. Mark Task as Complete
5. Plot Task Status
6. Quit
Choose an option (1-6): 4
```

Thank You