MODULE 1
CLASSES AND OBJECTS

NASA has task to create a Space Exploration Mission Planner represents a fascinating and instructive challenge. This project aims to simulate the complex operations involved in planning and executing space missions, albeit on a much-simplified scale. By focusing on the core components of space exploration—spacecraft, astronauts, and the missions themselves—this project provides a practical application of object-oriented programming (OOP) principles in Java, one of the most versatile and widely used programming languages in the world.

Java's OOP capabilities make it an ideal choice for modeling real-world entities and their interactions. In this project, we will design and implement three primary classes: Spacecraft, Astronaut, and SpaceMission. Each class will encapsulate specific attributes and behaviors relevant to its real-world counterpart, such as the spacecraft's type and capacity, the astronaut's rank and specialization, and the mission's duration and crew composition. By adhering to Java's OOP principles, such as encapsulation, inheritance, and polymorphism, we can create a modular and scalable system that could, in theory, be expanded into a more complex simulation.

The Spacecraft class will model various types of spacecraft, focusing on attributes like name, type (e.g., shuttle, satellite), speed, and capacity. The Astronaut class will represent individuals with specific roles and skills, necessary for the success of space missions. The SpaceMission class will tie these elements together, managing the logistics of space missions, including the assignment of spacecraft and crew.

To demonstrate the interactions between these classes and the functionality of our system, we will also create a Main class. This class will serve as the entry point for the simulation, using arrays of objects to manage and manipulate multiple spacecraft, astronauts, and missions simultaneously. Through this approach, we aim to illustrate not only the technical aspects of Java programming but also the thought processes behind modeling complex systems in a simplified manner.

Basic Questions

a. Implement the Spacecraft Class:

   Create a Spacecraft class with attributes for name, type (e.g., shuttle, satellite, cargo, crewed), speed, and capacity. Include a constructor to initialize these attributes and methods to get and set their values. Create one method named "launch" to print the spacecraft name.

b. Implement the Astronaut Class:

   Design an Astronaut class with attributes for name, rank (e.g., commander, pilot), and specialization (e.g., engineer, scientist). Provide a constructor for initializing these attributes and appropriate getter and setter methods. Create one method named "performEVA" to print the Astronaut's name.

c. Implement the SpaceMission Class:

Develop a SpaceMission class with attributes for mission name, duration (in days), assigned spacecraft, and crew (an array of Astronauts). Include methods to add or remove astronauts from the crew. Create one method to print the SpaceMission name and print all the signed all astronaut name.

Intermediate Questions

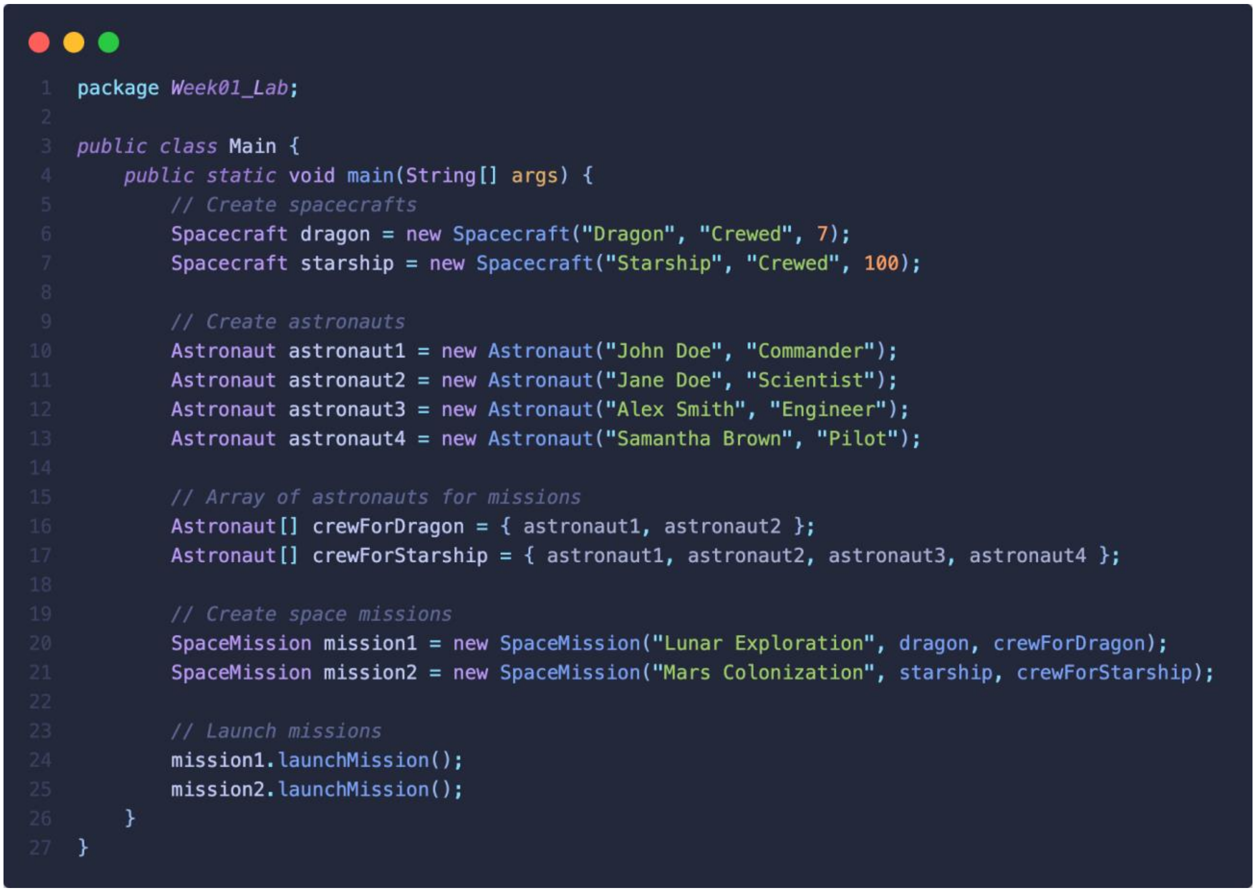
a. Array Management in SpaceMission:

Write a method in the SpaceMission class to add an astronaut to the mission, ensuring there's no duplication of astronauts. Implement another method to remove an astronaut by name.

b. Spacecraft Assignment:

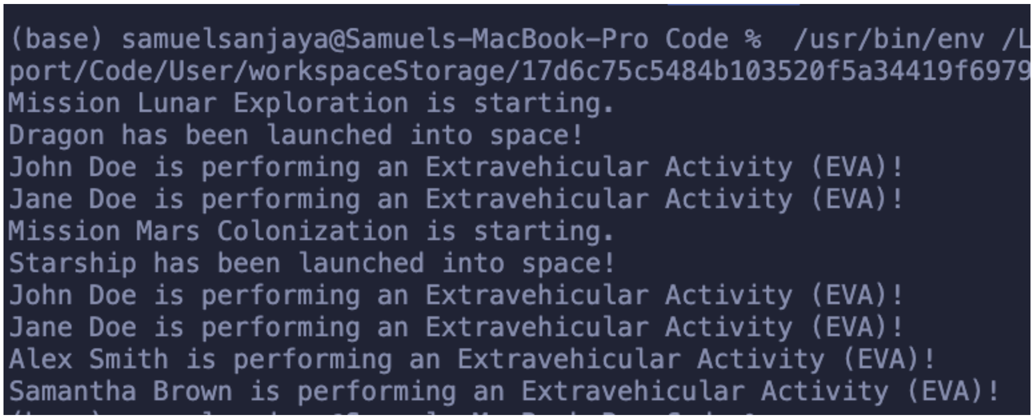
In the SpaceMission class, implement a method to assign a Spacecraft to the mission, ensuring that the spacecraft's capacity is not exceeded by the astronaut crew size.

Example of the Main Class:



```java
package Week01_Lab;

public class Main {
    public static void main(String[] args) {
        // Create spacecrafts
        Spacecraft dragon = new Spacecraft("Dragon", "Crewed", 7);
        Spacecraft starship = new Spacecraft("Starship", "Crewed", 100);

        // Create astronauts
        Astronaut astronaut1 = new Astronaut("John Doe", "Commander");
        Astronaut astronaut2 = new Astronaut("Jane Doe", "Scientist");
        Astronaut astronaut3 = new Astronaut("Alex Smith", "Engineer");
        Astronaut astronaut4 = new Astronaut("Samantha Brown", "Pilot");

        // Array of astronauts for missions
        Astronaut[] crewForDragon = { astronaut1, astronaut2 };
        Astronaut[] crewForStarship = { astronaut1, astronaut2, astronaut3, astronaut4 };

        // Create space missions
        SpaceMission mission1 = new SpaceMission("Lunar Exploration", dragon, crewForDragon);
        SpaceMission mission2 = new SpaceMission("Mars Colonization", starship, crewForStarship);

        // Launch missions
        mission1.launchMission();
        mission2.launchMission();
    }
}
```

Expected Output:



```
(base) samuelsanjaya@Samuels-MacBook-Pro Code %  /usr/bin/env /L
port/Code/User/workspaceStorage/17d6c75c5484b103520f5a34419f6979
Mission Lunar Exploration is starting.
Dragon has been launched into space!
John Doe is performing an Extravehicular Activity (EVA)!
Jane Doe is performing an Extravehicular Activity (EVA)!
Mission Mars Colonization is starting.
Starship has been launched into space!
John Doe is performing an Extravehicular Activity (EVA)!
Jane Doe is performing an Extravehicular Activity (EVA)!
Alex Smith is performing an Extravehicular Activity (EVA)!
Samantha Brown is performing an Extravehicular Activity (EVA)!
```