## 

## 

**Experiment 3:**

**Object Oriented Design and Implementation**

CPE106L (Software Design Laboratory)

**Member 1: Cyris Ken M. Alipio**

**Member 2: Neil Emmanuel Macaro**

**Member 3: Paolo Sarmiento**

Group No.: **6**

Section: **B2**

## **PreLab**



|  |
| --- |
| **Readings, Insights, and Reflection**  **Systems Analysis and Design: An Object Oriented Approach with UML**  **9781119561217**  Alipio, Sarmiento, Macaro  In our exploration of Chapters 4 and 5 across various resources, we've delved into essential techniques and tools for visualizing software systems using UML diagrams. Chapter 4 focuses on creating Purpose Case Charts and Use Case Diagrams, which are crucial for understanding system interactions and functionalities. These diagrams allow us to capture the connections between actors and the system, providing insights into user interactions and system behaviors. Moving on to Chapter 5, we've learned about Class Diagrams, which are essential for representing the static structure of the system, including class modeling, associations, and advanced UML concepts like aggregation and composition. These chapters have equipped us with practical insights and tools for effectively modeling and designing software systems, while also fostering our skill development in object-oriented programming and system analysis. Through straightforward explanations, informative examples, and UML-based techniques, they have become invaluable resources for us as we navigate the complexities of software design processes.  **Fundamentals of Python: First Programs**  **9781337671019**  Alipio, Sarmiento, Macaro In Chapter 9 of our study material, we're diving into how to design software using objects. We're learning about important concepts like class inheritance and polymorphism in Python, which help us make software that's easy to maintain and reuse. By working on projects where we create diagrams called UML diagrams, we're getting hands-on practice applying these concepts. This chapter also talks about planning carefully during the software development process and discusses things like refining diagrams and making sure our software meets all the requirements. Overall, Chapter 9 is teaching us how to design software that's organized and easy to work with, which is important for making sure our projects run smoothly. |

**Answers to Questions**

1. a. Is owned by a particular instance of a class and no other

2. c. self

3. b. set the instance variables to initial values

4. b. always must have at least one parameter name, called self

5. b. the entire class in which it is introduced

6. b. when it can no longer be referenced anywhere in a program

7. a. all instances of a class have in common

8. b. A.\_\_init\_\_(self)

9. b. pickle them using the pickle function dump

10. a. has a single header but different bodies in different classes