Answers and UML Diagram

* **What type of a design pattern (creational, structural, behavioural, custom) did you employ in this assignment? Why?**

For this assignment, we used the adapter pattern, which is a structural design pattern. This is because we had a simulator that assumes a similar function call for all the vehicles; hence, the need to make the corresponding functions for the Boat, Car, and Airplane adapt the legacy code to the travel simulator.

* **Describe a software design pattern that you have used before this course, even if you were not aware of it.**

We used the factory design pattern in the Assignment1(Card Game) for SEP200, where you have a card game Factory that generates 2 types of Players (Interface), YouthPlayer and AdultPlayer (Concrete Classes).

* **Do you feel standard design patterns are of great assistance, or great hindrance, in software design? Give advantages and disadvantages.**

We think that design patterns are of great assistance, but they have their disadvantages as well.

Advantages of implementing design patterns are:

* + Utilizing various design patterns tailored to specific problem types not only expedites the resolution of complex issues but also tends to yield an optimal solution.
  + Using pre-existing design patterns enables the creation of reusable solutions applicable to various scenarios and cases.
  + Design patterns, like the abstract factory pattern, facilitate scalable system expansion and feature addition without major codebase modifications, averting potential bugs with new feature introductions.

Disadvantages of implementing design patterns are:

* + Using a specific design pattern may result in a rigid code structure, limiting flexibility. This could pose an issue when the project's business requirements evolve over time.
  + Complex problems may require combinations of design patterns; a project could use more than three patterns at a time, leading to increased complexity.
* **What are some major drawbacks of the Singleton design pattern?**

There are some major drawbacks for the singleton pattern:

* + Using the singleton pattern in a multi-threaded application can result in concurrent access issues, potentially causing race conditions or synchronization problems.
  + The Singleton class, offering a global state, allows unrestricted access and modification from any part of the program. This can potentially result in unintended bugs and logical errors, especially when managing database connections.
  + Extending or subclassing a singleton class is challenging due to its inherent restriction to a single instance.

(*UML diagram below*)**UML Diagram**

