**Swinburne University of Technology**

**Name: Nguyen Manh Dung**

**Id: 104181789**

**Task 2 OOP Midterm Test**

**Question 1:**

In oriented object programming, the polymorphism principle enables objects from various classes to be viewed as objects of the base class. This mean that even a class common interface may be used to refer and interact with all derived classes, even if each have its own implementation.

In Task 1, polymorphism principle is being demonstrated through the power to group different strategies, such as “MinMax” and Average”referred to same base class. Though each strategy has its unique behavior, they can be accessed and changed by utilizing the base class methods and properties, which is SummaryStrategy.

**Question 2:**

Programs created using OOP often consist of large code bases with interconnected components. As a result, managing the complexity of updating several code files is a difficult process. Abstraction is the answer to this. Basically, the idea is to hide unnecessary information and just look at an object's key characteristics. For example, in above program, the abstract class SummaryStrategt defines all the properties and methods for all strategies which let all strategies to inherit from it.

**Question 3:**

The issue with the original design in task 1 is that it did not implement the base class SummaryStrategy, which will make \_strategy fields in DataAnalyser class hard to handle different summary approaches if we update the project further. Thus, the original design did not follow abstraction for the design.

UML Diagram:

