Semester Test

Task 2

Thanh Minh

1. **Describe the principle of polymorphism and how it was used in Task 1**
   Polymorphism refers to a variety of ways to perform a task. Additionally, it also give the child classes to inherited from their parents but can be in different form. As an example in Task 1, polymorphism was implemented for creating the new abstract class SummaryStrategy which will support for the AverageSummary and MinMaxSummary classes. This method will help us to write the code more efficient.

2. **Using an example, explain the principle of abstraction.**
   An example is that when we use out laptop, we have interacted with its interface which are keyboards, screen … while the interior or its implementation are hidden which are mother board, chips, CPU,… through out this example, we can understand about the principle of abstraction is that, it only show the necessary details to the users, developers and hiding the rest of it.

3. **What was the issue with the original design in Task 1? Consider what would happen if we had 50 different summary approaches to choose from instead of just 2.**
   The error in the original design in task 1 is that the Summarise method is dependent too much on the string parameter it takes, which require the users to remember all the key input for selecting the right type of strategy. If there are 50 different summary approaches, there will be a lot difficult for the users to remember all 50 keywords to select the right strategy. After implementing the polymorphism and abstraction (after Task 1) the code has been more flexible and can be extended in the future easier and can accommodate with a lot of summary strategies.