

1. Describe the principle of polymorphism and how it was used in Task 1

Polymorphism is a concept of reusing code and expanding the program without impacting the original code. It allows developers to create various classes by recycling the same interface, which saves a lot of time and easy to debug. This concept was used in Task 1 to decrease the amount of repetitive code and the amount of memory space used to run the program.

2. Using an example, explain the principle of abstraction. In your answer, refer to how classes in OO programs are designed.

Abstraction is one of the four key concepts in OOP, with the sole purpose of hiding unnecessary information from the user and only show the important attributes of the program. For example, instead of calling the functions such as Average in Average Summary class and Minimum and Maximum in MinMax strategy. This put program at risk of revealing the internal classes and data of the program. Instead of waiting for those two possible "if" conditions in Summarise function, DataAnalyser class would use the abstract class to call for the PrintSummary function that both MinMaxSummary and AverageSummary classes contain only once. This hides the necessary functions from both classes to perform calculations and return the summarised results.

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 main issue with the original design in Task 1 was the possibility of creating repetitive code that would be redundant for the program, affecting its performance and efficiency. It is evident in the original design where it requires two variables for containing each type of summary methods to access the functions that are responsible for calculation. If we had 50 different summary approaches, the DataAnalyser class would contain 50 different variables for each type of summary strategy, as well as 50 "if" conditions within the Summarise function. Therefore, programs that implement the original design in Task 1 would inefficient, taking up a huge amount of memory space, and potentially have extremely poor performance.