1.

The core idea of polymorphism in object-oriented programming (OOP) enables objects of various kinds to be handled as instances of a single superclass or interface. It promotes flexibility and extensibility in software design by enabling the writing of code in a generic and reusable way.

In the first task, polymorphism is used by creating a parent class called SummaryStrategy and 2 children classes called AverageSummary and MinMaxSummary.

2.

We can consider an automobile, for example, to be an abstraction of a physical item. We don't need to comprehend every nuance of the electrical system's intricacy, the mechanics of the gearbox, or the engine's operation when we engage with an automobile. As an alternative, we concentrate on the high-level ideas and procedures that are important to us, such as starting the engine, accelerating, braking, and steering.

3.

If the first design was kept, it would be really challenging to extend the program further when another SummaryStrategy is added. The code in the DataAnalyser will need to be updated every single time a new SummaryStrategy is added. If there are 50 different summary approaches, the DataAnalyser will be packed with too much data as well as too complex code.