– Semester Test –

Task 2

1.

a, Describe the principle of polymorphism

Polymorphism is a programming concept that refers to the ability of a variable, function or object to take on multiple forms. It means that a single method can have different implementations, based on the object type that it is called on.

b, How it was used in Task 1

During the Task 1, I use Polymorphism to implement the abstract class “SummaryStrategy” along with the “AverageSummary” and “MinMaxSummary” classes, which are inherit from the Summary one. The “DataAnalyser” class is using the “SummaryStrategy” as its base class as well. This means that the “DataAnalyser” class can use any of the methods defined in the “SummaryStrategy” class.

One of these methods is the “PrintSummary” method, so that they also have access to the “PrintSummary” method. This makes it easier to print out the values in “DataAnalyser”, since you can call the “PrintSummary” method on an object of both the “AverageSummary” and “MinMaxSummary” class.

2. Using an example, explain the principle of abstraction

Abstraction is the process of resolving or obscuring technical details while emphasizing key qualities in order to represent complicated real-world objects. These objects, which may take the form of classes, methods, or interfaces, enable us to develop a higher-level viewpoint on the issue or system at hand. Usually, interfaces or abstract classes are used in programming to achieve abstraction.

A car, for instance, includes a number of intricate parts, including an engine, transmission, suspension, steering, braking system, etc. These elements are significant considerations for various applications (like driving). These specifics, however, are less crucial when developing a website for a vehicle dealership.

We can abstract away much of the implementation specifics of the car by building a simplified "Car" object that merely displays the make, model, and color. As a result, working with the "Car" object is much simpler because we can concentrate on its key characteristics and behavior without having to worry about its underlying complexity.

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 Task 1 design had a major flaw in that it chose which sort of summary technique to use based on string parameters. As additional summary approaches were incorporated into the system, this made it very challenging to maintain. To examine the value of the parameter and choose the appropriate summary method in the case where there were 50 or more distinct summary approaches, it would be necessary to add 50 different conditional statements.

4. References:

Available at:
https://www.cs.sjsu.edu/~pearce/modules/lectures/ood/principles/Abstraction.htm