OOP 8.1 Task 2

# Ekrar Uddin Mohammed Efaz

## 103494172

1. Polymorphism allows child classes to use abstract methods from a parent class by completely overwriting their function for the specific child class or adding additional functionality.

For Example: In our task 1 our parent class *SummaryStrategy* has an abstract method *PrintSummary().* The child classes *AverageSummary* and *MinMaxSummary* can use this *PrintSummary()* method with their own functionalities and when called upon will execute the function corresponding to the class it was called upon.

1. Abstraction is the OOP principle that only shows essential information to the users hiding the inner workings of the program.Text

   Description automatically generated

For Example: In task1 our AverageSummary class has a private method Average() which finds the average of the given list of numbers. But the user using our Program class doesn’t need to know how the average is being calculated, he just needs to provide a list of numbers and call the method PrintSummary for the AverageSummary class to get the average. Here because of abstraction the user is not exposed to the inner workings of how the average is determined.

1. The original program didn’t use OOP principles which would lead to design problems if the program went on to add more Summary approaches. We had to add a new if-else statement every time we updated a new Summary approach. The original program design had major flaws in case the program had 50 options of Summary Strategy we had to write 50 different if-else statements for the program to output the correct Summary.