Examiner Report	
Qualification Name	Higher Education Qualification
Qualification Level	Diploma
Date/ Series	April 2024
Module	Object Oriented Programming
Question no.	**comments**
A1	The majority of candidates answered this question, with most being able to discuss where the object-oriented paradigm might not be the most appropriate paradigm. Speed and the complexity of the code were the most common answers, though not everyone described a real-world practical scenario, so missed the opportunity to gain full marks. Instead of a scenario, some candidates produced a lot of code with no explanation, which did not answer the question. For part b, most candidates could describe three different object-oriented concepts, though to gain a high mark they needed to be related to promoting reuse. Inheritance and polymorphism were common answers, though it was not enough just to describe what these were, to gain a high mark the candidate needed to discuss them with respect to how they can help promote code reuse. In some cases, candidates implied that using the method name again, as in the concepts of overriding and overloading was the same as reusing code.
Question no.	**Comments**
A2	This was a popular question, with most candidates able to describe what is meant by the term abstract class in part a. Where candidates lost marks was in not providing a real-world practical scenario to show where it might be used too. A number of candidates produced code to show the definition of an abstract class, but to gain a high mark they needed to describe its purpose. In some cases, candidates discussed the concept of abstraction and data hiding rather than abstract classes. For Part b some candidate just described what inheritance means, listing every type they could think of, without discussing them in the context of hierarchical and hybrid inheritance. Code fragments were asked for to illustrate the answers, which did not mean producing a lot of code with no explanation of how they differed. Some candidates included UML diagrams instead of code, some

	credit was given for these provided they contributed to the discussion on how the two types of inheritance differed.
Question no.	**Comments**
A3	This question was less popular, though had a higher pass rate, with a number of candidates getting full marks for part a. Some candidates could not fully explain what SOLID stood for, often not remembering the Interface Segregation Principle and the Dependency Inversion Principle. In part b, the two most popular choices were Single Responsibility Principle and the Open Closed Principle. For the Single Responsibility some candidates assumed the class could only have one method, rather than one responsibility. Overall, most candidates could say what the principle was, but were weaker on saying why it was important. To gain a higher mark, code examples were required to show examples of the principle being violated or adhered to, which some candidates failed to do, or could only provide code for one type.

Question no.	**comments**
B4	In part a) a significant number of candidates did not provide valid OCL, instead providing code or pseudocode. In other cases, valid OCL was provided and described, but the part of the question asking how OCL may be used to improve quality was neglected. In part b) most answers correctly distinguished between coupling and cohesion. Some candidates merely stated that low coupling and high cohesion are desirable, but did not elaborate upon this to demonstrate an understanding of what this means in practice.
Question no.	**comments**
B5	In most cases, a class diagram representing the scenario was provided – although in a few cases, a use case diagram was provided instead. Although most identified the classes, the potential for inheritance (for instance that car and van are specialised cases of a base class vehicle) was not always exploited. Data types were also often incorrect, for instance using strings to hold numeric values or vice versa. In part b), some candidates only answered part of the question, for instance discriminating between private, public and protected and not describing the three compartments of a class, or vice versa.
Question no.	**comments**
B6	In part a), there was quite widespread confusion as to what exactly is meant by procedural and structured language. In particular, quite a number of candidates believed that structured language meant something entirely different, suggesting that spending some time engaging with these concepts is needed.

	In part b), four items of design pattern documentation were requested. In some cases, the items listed were not recognisable as items of design pattern documentation. In other cases, they were named but the descriptions suggested an incomplete understanding, suggesting some rote memorisation without proper engagement.