Examiner Report	
Qualification Name	Higher Education Qualification
Qualification Level	Diploma
Date/ Series	April 2026
Module	Object Oriented Programming

Question no.	**comments**
A1	This question was attempted only by a few candidates and they achieved high marks. Candidates who understood the Object Constraint Language (OCL) were able to produce appropriate code for the different scenarios. Marks were lost due to minor syntax omissions for the OCL in Part A.
Question no.	**Comments**
A2	Two-thirds of the candidates attempted this question. For Part A some candidates spent time saying what polymorphism was in general and also different aspects of inheritance, rather than addressing what ad-hoc and parametric polymorphism entailed. Whilst ad-hoc polymorphism can use method and operator overloading, some candidates mistakenly assumed parametric polymorphism was method overriding. Marks were lost by not including examples of code too. Part B was poorly answered, with answers mostly just saying one approach was easier than the other for the three aspects, without giving any reasons why.
Question no.	**Comments**
A3	Over 85% of candidates attempted this question and over 90% achieve a pass grade. In Part A most candidates could state what the different SOLID aspects were. Marks were mostly lost by not knowing what the “ID” stood for - Interface Segregation and Dependency Inversion Principles. Some candidates lost marks by just saying what the letters stood for, without any further explanation.

	For Part B, most candidates could describe what black and white box testing was and could provide some advantages and disadvantages. Marks were mostly lost by not providing enough detail, or giving very basic answers, such as saying black box testing was easy and white box hard. Most marks were lost in b(iii) where candidates were not able to determine where white and box testing should be used in the software life cycle, with some giving examples of types of applications instead.

Question no.	**comments**
B4	Part a) - This question attracted a range of answers. In most cases a use-case diagram of reasonable clarity was presented, even if, in some cases, it was confused or overcomplicated. In some cases, a class diagram was given instead. In others, features of the use case diagram were missing or misrepresented, such as actors and the system boundary. Part b) - In this question, most candidates correctly suggested that use case diagrams help the developer focus on ensuring core functionalities are realised. That they serve as a good tool for interacting with clients due to their relative ease of understanding compared to more technical diagrams and can also serve in testing phase to ensure these core functionalities are working. However, some candidates also incorrectly asserted that these diagrams reduce coding errors.
Question no.	**comments**
B5	This was the second most attempted question in this year's exam. Part a) - Many candidates correctly identified procedural and structured programming paradigms as having contributed to the object-oriented paradigm. Some candidates were unfamiliar with these terms and instead described the features of the object-oriented paradigm without meaningfully linking back to their antecedents in earlier approaches. Part b) - Many candidates were familiar with the terms coupling and cohesion and correctly asserted that is typically our objective to aim for low coupling and high cohesion. In the final part of the question, it was not always made clear how encapsulation helps us to realise this objective. In some cases, candidates did not appear to be familiar enough with these terms to articulate their meaning or relate them to encapsulation.
Question no.	**comments**
B6	Part a) - There were very few viable class diagrams submitted in response to this question. In some cases, the diagram was not a UML class diagram at all. In others, member visibility symbols,

	member data types, and relationships between classes were omitted. In most cases, the class variable was not correctly shown. Part b) - Few candidates used an appropriate form of diagram to answer this question. It was also common not to identify which of the two diagrams was proposed to be “valid” vs “invalid”. Part c) - This question was relatively well answered, with most candidates able to explain the difference between typed and untyped languages. Fewer candidates correctly identified examples of languages falling into these categories.