BCS Higher Education Qualification

Diploma

April 2025

EXAMINERS' REPORT

Object Oriented Programming

Questions Report:

A1	
	This question was answered by the majority of the candidates, with over 50% passing, some providing excellent answers, gaining full marks. For part a) most candidates could describe the different types of inheritance, though not everyone provided code, which was required for full marks. In some instances, multilevel and hierarchical inheritance were mixed up, or multilevel and multiple inheritance. Part b) was not always attempted, or some candidates discussed encapsulation or polymorphism, which was not appropriate when looking at other types of relationships. In some cases, the examples provided were types of inheritance already seen in part a).
A2	
	A less popular question, with 50% attempting it, though over 70% passed. For part a) most candidates knew what the SOLID principles were, marks were lost by either not giving an explanation, or it was too brief. The Integration Segregation Principle was the one a number of candidates could not remember. Part b) was not always attempted. Weaker answers would only look at Abstract Data Types (ADT) and not mention classes. Others discussed what procedural programming was, which was not required.
A3	
	This question was answered by most of the candidates, with 80% passing. Most candidates could describe what white and black box testing was in part a). Higher marks went to candidates that tailored the answer to look at these testing approaches from an object-oriented programming point of view. Marks were mostly lost where candidates mixed up the two approaches. Marks were lost when discussing the advantages and disadvantages by providing very brief bullet points. Better answers made distinct points for the advantages and disadvantages, rather than stating the opposite for each approach, for example, saying black box testing was easy and white box testing was hard. Not many candidates attempted part b) or provided very brief answers.
B4	

	This question was attempted by three quarters of this year’s candidates, of whom 60% scored a pass mark. In part a), most candidates gave a good description of object-oriented paradigm, but many were unsure of the nature of procedural and structured paradigms, and how they relate to each other and to OOP, leading to some confused answers. In part b), many candidates were able to identify typed and untyped languages and seemed to have a basic idea what the difference between these approaches is. However, there was some misconception that untyped languages don’t have variable data types, rather than that data type is inferred from context.
B5	
	This question was attempted by around one half of this year’s candidates, of which 50% scored a pass mark. In part a), it was apparent that some candidates may not have studied design patterns, despite their explicit mention in the syllabus, giving answers that referred to design more generally or to UML. In other answers, design patterns were incorrectly given as examples in the wrong category. In part b), which was concerned with how UML diagrams can be used to support testing, most candidates correctly identified use case diagrams as source of test scenarios relating to what the software should offer to users. The second required example was more varied, with many answers being rather vague and relating to class diagrams.
B6	
	This question was attempted by two thirds of this year’s candidates, of which 65% scored a pass mark. In part a) there was much variability. In some cases, candidates give use case diagram instead of a class diagram. In many cases, the opportunity for a generalisation using borrower and librarian was missed. In many cases, the cardinality limitation that enforces the limit of borrowing 4 items was missed. In many cases, data members were inexplicably set to public. In some cases, connections between classes were muddled, such as including inappropriate aggregations or compositions. However, most candidates were able to score some points, with appropriate classes that were at least in part appropriately interconnected, with name, attributes and behaviours placed in the right regions of the box. In part b), the question asks for an interaction diagram. Valid interaction diagrams would be sequence diagrams or collaboration (communication) diagrams. Many incorrectly gave an object diagram, which are (structural rather than behavioural). However, given that the question mentions instantiation of classes, it is somewhat ambiguous and could be misinterpreted as requesting an object diagram. This is exacerbated because object interaction diagram is not standard UML terminology and appears to conflate two different concepts Given this, in some cases, some marks were awarded for answers featuring object diagrams.

	In part c), most identified the three regions in a class diagram box and gave the notation for private, public and protected. Not all candidates related this to the class diagram given in part a), losing some of the available marks.