BCS Higher Education Qualification

Diploma

April 2023

EXAMINERS' REPORT

Object Oriented Programming

Questions Report:

A1	
	Candidates were able to score better marks on part a), where they were able to show they understood what public visibility in a class meant. Weaker answers described the three types of visibility, namely public, protected and private, but then failed to give an example of where public visibility could be used in a real-world scenario. It is important that candidates tailor the examples to the question asked, in this case the scenario was about variables and public visibility, rather than classes or methods. Part b) was often not attempted, or the answers lacked sufficient detail to score a high mark. A number of candidates mixed up inheritance with nested classes, so did not gain any credit. Good answers showed code that demonstrated a nested class within another class, with higher marks given to answers that showed some detail in the classes, for example, added sample attributes and methods, rather than basic outline code.
A2	
	Part a) was poorly answered with a lot of candidates stating that a first-class object was the first object to be instantiated in a system. Others likened them to the singleton pattern. To achieve a good mark, candidates needed to describe what a first-class object was, giving an appropriate example. Part b) was answered better, though a number of candidates simply described what multiple inheritance was or confused it with multi-level inheritance. To achieve a good mark, most of the discussion should be on what issues multi-inheritance might cause, such as method overloading. It is acceptable to include some background explanation of what multi-inheritance is but it should not be the sole answer. Answers should also include some example code to illustrate the chosen issue.
A3	
	For Part a), most candidates could describe what private visibility is. Most marks were lost where answers implied that private meant information security, which was not appropriate in this context. It is important to read the question carefully, which was about why a method should be set to private visibility, rather than the attributes of the class. A large number of candidates did not attempt Part b) of the question. Good answers described what the Liskov Substitution Principle was, then provided code that defined a supertype and one or two subtypes, with a demonstration of how a subtype object could be used to substitute an object of the supertype.

	Weaker answers included code to show a supertype and subtype but failed to demonstrate the Liskov Substitution Principle. Others mixed up the principle with either method overloading or overriding, which was not appropriate.
B4	
	In part a), most answers listed and briefly described prominent features in a use-case diagram, and many also described some less prominent features. In most cases, an example diagram was provided that was annotated to show where each feature was to be found. In a few cases, candidates spoke about class diagrams rather than use-case diagrams. The question also asked for a description of how these diagrams aid the object-oriented development process. This part of the question was less clearly answered, with some erroneous comments about converting the use-cases into classes. Most candidates correctly asserted that a key benefit of this diagram type is its understandability to stakeholders as well as the development team. In part b), candidates were asked to describe how object-oriented code can be tested, and how use-case diagrams might be used in this process. Most candidates described a number of broad testing methods (most commonly black box, white box, unit, and integration). Fewer made a fully convincing argument describing how the use-case diagram might be used to identify scenarios that might be tested to confirm correct operation of the application, and that it fits the requirements.
B5	
	This question was somewhat straightforward, requiring that the candidate describe some key terms in object-oriented programming, which were presented in dual form (e.g., abstract vs concrete classes). Candidates were rewarded for knowing these terms, and the question scored the second highest average mark in the exam. Of the answers presented, the most common errors were the conflation of generalisation and specialisation, and some complicated answers describing encapsulation and data hiding. The strongest answered part was v), which asked candidates to describe classes and objects. Where full marks were not achieved, this was often due to lack of adherence to the part of the question that asked for real-world scenarios in each description. In many cases, only an abstract theoretical definition was given without accompanying real-world application(s).
B6	
	In part a), candidates were asked to examine a class diagram and then assess whether five object diagrams were valid or invalid, explaining why for each. Few candidates correctly evaluated all five diagrams, although most were able to identify a number of inconsistencies relative to the class diagram. The main reason for the relatively poor performance in this question, may have been that part b) was not fully understood. Part b) required that the two-class diagram provided, was to be implemented in code. It included a class variable, and required constructors, and some functions (e.g., to increment the class variable). Where answers were attempted, in many cases, skeletal classes were provided that lacked these features, or had inappropriate member visibilities.