BCS Higher Education Qualification

Diploma

October 2022

EXAMINERS' REPORT

Object Oriented Programming

General comments[1]

Questions Report:

A1	Syllabus Content: 1.2; 2.3
	This question was answered by over 50% of candidates, but had a low pass rate. Candidates were able to score good marks on Part A, where they were able to show they understood what protected visibility in a class meant. Weaker answers described the three types of visibility, but then failed to give an example of where protected could be used in a real-world scenario. Most marks were lost where answers implied that protected meant information security, which was not appropriate in this context. Part B was often not attempted, or the answers lacked sufficient detail to score a high mark. A number of candidates provided general constructors only, which showed no evidence of any conversion. Some examples showed type conversion, few provided answers that converted an object passed as a parameter to the type of the conversion constructor's class. A small minority of candidates mixed up conversion constructors with overloading.
A2	Syllabus Content: 2.2, 2.3
	This question was answers by over 64% of candidates and most candidates scored well in part A. Better answers compared the advantages and disadvantages of manual and automatic memory management, with relevant examples given from object-oriented programming. Weaker answers listed a set of bullet points without going into any depth, such as stating a disadvantage was "high cost", without explaining what they meant by this, which was needed for a higher mark. The advantages given were often just the opposite to the disadvantage, such as "low cost". A substantial number of candidates did not attempt Part B of the question. The more able candidates were able to provide code to illustrate hybrid inheritance, often including a class diagram to explain how the classes demonstrated this. Weaker answers mixed up hybrid inheritance with multiple inheritance and some included code that only demonstrated one type, typically single inheritance. Some candidates struggled to suggest a real-world example, though credit was given if they could outline what hybrid inheritance would look like using Class A, B, C type examples.

A3	Syllabus Content: 3.4; 4.2
	This question was attempted by over 50% of the candidates and most scored good marks on Part A of this question. The majority of candidates were able to describe what a singleton class was, though some did not get beyond stating it was a single instance of a class. To obtain a high mark, an appropriate real-world example was needed, which some candidates were not able to provide. In Part B, most candidates could explain what an is-a inter-class relationship was and provided appropriate code to demonstrate this. Less able candidates were unable to provide a suitable has-a inter-class relationship, or mixed the two types up. A good answer needed to clearly show inheritance for an is-a relationship and composition for a has-a relationship, sometimes, the answer showed two classes, but not any code to show how the inter-class relationship would work.
B4	Syllabus Content: 3.2; 4.4
	Question B4 was attempted by almost 80% of candidates, making it the most popular question on this year's paper. The quality of answers varied enormously. In part (a) the most common mistake was not concentrating on structural constraints, but (in extreme cases) simply listing all of the class names, along with the operations and variables they contain. Many of the answers were excessively long and unfocussed. Many candidates did correctly identify the multiplicity of the inter-class relationships, although other structural constraints were quite vaguely or confusingly expressed. In part (b), many candidates simply listed, with brief explanations, one or two testing methods (such as black box, white box, or unit testing), ignoring the part of the question that asked that the tests could be used on the class diagram from part (a). Some marks were awarded if some testing methods were accurately described. Higher marks were only given if the candidate linked the testing method to object oriented programming and the class diagram specifically.
B5	Syllabus Content: 3.2; 3.4
	This question was attempted by approaching 65% of candidates, but was one of the two least well answered questions on this year's paper. Part (a) asked that candidates identify an example design pattern in one of three categories (creational, structural, behavioural), and then go on to describe it, including the problem it addresses and the basis of its offered. In many cases, candidates proposed design patterns that did not match the categories (e.g., an iterator being identified as creational). Other issues included not providing any description of the design patterns identified, or not clearly stating the problem they are intended to address. In part (b), candidates were asked about the use of object interaction and object state transition diagrams. Example diagrams were not requested, but most candidates attempting this question provided them, perhaps to help answer the part of the question asking for a real-world scenario. It was a common mistake to provide/describe an object diagram instead of an object interaction diagram. In some cases, the proposed states in the state transition diagram examples were not actually states (for instance, some conveyed actions rather than states).
B6	Syllabus Content: 2.3, 3.3; 4.4
	Question B6 was the second most popular question on this year's paper, with nearly 80% of candidates attempting it. Approximately half of those attempting the question gave a good answer.

	Part (a) asked about the key elements of object constraint language (OCL), for which most candidates were able to speak about pre and post conditions and context (with some also mentioning invariants and guards), and the role of OCL to formalise aspects of a UML diagram that are inherently lacking in detail. In part (b), most candidates seemed to largely follow the OCL statement, but descriptions were rarely completely clear, and several misinterpreted the statement (e.g., believing that the mark argument was added to the grade variable rather than replacing it, or that the mark had to be >100). Part (c) was generally quite poorly answered, with most candidates rephrasing the same point over and over again rather than truly identifying distinct and meaningful advantages/disadvantages of getters/setters.