BCS Higher Education Qualification

Diploma

March 2020

EXAMINERS' REPORT

Object Oriented Programming

General comments
The number of candidates attempting this paper was greatly reduced due to the Covid-19 pandemic. As a consequence, the overall statistics may have no meaning. However, the paper pass rate on this occasion was similar to previous sittings.

Question number: A1
Syllabus area: Foundations parts 1.1,1.2, 1.3, 1.4
Total marks allocated: 25
Examiners' Guidance Notes
It is relatively difficult to comment on the breadth of answers for this paper as there were fewer candidates than normal as a consequence of the global Covid-19 pandemic. However, of those candidates that did sit the paper, this question was both the most popular and attracted the highest marks. Most candidates were able to define the key terms in part (a) reasonably clearly, and were able to offer some insight into the differences between procedural and structured programming and how these paradigms contributed to the development of object oriented programming in part (b).

Question number: A2
Syllabus area: Design part 3.4, Practice part 4.4
Total marks allocated: 25
Examiners' Guidance Notes
This question was attempted by half of the candidates sitting this year's paper. In part (a), most candidates were able to identify some but not all of the concepts mentioned, or gave rather vague answers. There were some very good answers explaining the role of OCL in object oriented development, part (b). The question of testing, (c), attracted a wide variety of answers, but very little detail was included in most. In particular, where white box testing was identified, it was not clearly explained. It would have been nice to see some more OO-specific testing proposals mentioned, such as mock objects and creating harnesses to validate classes/objects in isolation.

Question number: A3
Syllabus area: Design part 3.1
Total marks allocated: 25
Examiners’ Guidance Notes
This question was attempted by three-quarters of this year’s candidates. Some candidates mistakenly drew a class diagram rather than a use case diagram in part (a). For many others, the diagram was incomplete or contained structures that violates the guidelines associated with use-cases. Most candidates identified some appropriate uses of use case diagrams. Overall, this question had the second best average mark

Question number: B4
Syllabus area: Foundations part 1.2, Concepts part 2.1, 2.2 and Practice part 4.3.
Total marks allocated: 25
Examiners’ Guidance Notes
This was the least well answered question in this year’s paper, attracting the lowest average mark. A common mistake in part (a) was to confuse the concept of the Abstract Data Type with that of an abstract class. In part (b), many candidates wasted time writing full programs (such as main methods) to test objects, but did not follow the question carefully and construct the hierarchy requested – for instance, the hierarchy did not incorporate both is-a and has-a inter-class relationships.

Question number: B5
Syllabus area: Concepts part 2.3, Practice part 4.2
Total marks allocated: 25
Examiners’ Guidance Notes
This question was attempted by half of this sitting’s candidates. Many candidates were able to offer some insights as to the role of the garbage collector in question (a), but other answers lacked detail/clarity. Question (b), on the topic of code refactoring had a wide variety of answers, some of which appeared to be improvised/rely upon intuition rather than reflect true knowledge/understanding of this process.

Question number: B6
Syllabus area: Concepts) part 2.3, Practice part 4.2 and 4.3.
Total marks allocated: 25
Examiners’ Guidance Notes
Answers to part (a) were relatively good, with most candidates able to describe some of the information that can be gleaned from a method signature. Some candidates realised that some languages do not consider the return type to be part of the signature, but others did not. Since there is debate on this issue, if candidates included the claim that the return type is something that can be learned from examining a method signature this was not heavily penalised. In part (b), most candidates were able to show a code fragment that implements inheritance, and many were able to show composition. What was lacking in many answers was a convincing third approach to connecting classes.