BCS Higher Education Qualification

Diploma

November 2020

EXAMINERS' REPORT

Object-Oriented Programming

General comments

The number of candidates attempting this paper was significantly lower than for pre-pandemic offerings. Nevertheless, the average mark and pass rate for the paper was broadly in line with previous sittings. Candidates with practical experience of object oriented programming appeared to do better than those with only limited use of OO languages. The examiners' comments on question B6 highlight a common mistake made by candidates sitting this paper: a failure to read and follow the guidance given in the questions.

Questions Report:

A1	
	Syllabus areas: Section 1 (Foundations) and Section 2 (Concepts)
	Total Marks Allocated: 25
	Examiners' Guidance Notes
	All candidates attempted this question and the question was mostly answered well. The majority of candidates were able to describe five distinct features of the object-oriented paradigm. Full marks were given to answers that included an appropriate example of the feature and also explained why it was important, though this last part was often omitted. Weaker answers often just listed five features, with very little detail of what they meant. In some cases candidates gave identical answers for different concepts, in particular encapsulation and abstraction, neglecting to distinguish the difference between them.

A2	
	Syllabus area: Section 3 (Design) Part 3.4
	Total Marks Allocated: 25
	Examiners' Guidance Notes
a)	More than 70% attempted this question and for part a, good answers were produced to explain what the term design pattern meant for object-oriented development. The answers were more variable in explaining the motivation for using them from a programmer's point of view.
b)	For part b, most candidates could give a general description of three design patterns, with the adapter, decorator and singleton patterns being the most popular. Marks were lost for not stating the problem they address and how they might offer a solution. Good answers included an example of how they might be used in practice. A few candidates only attempted one or two design patterns, whilst others attempted more than three. In the latter case marks were only counted for the three best patterns.

A3	
	Syllabus area: Section 3 (Design) Parts 3.1, 3.2 and 3.3
	Total Marks Allocated: 25
	Examiners’ Guidance Notes
a)	More than 75% attempted this question and good answers were produced overall for part a). Most candidates identified the correct classes and how they related to each other. Marks were lost mostly for not identifying that some of the classes had similar properties, so that there was scope for inheritance. The question included attributes and methods for the classes, most answers included the correct attributes, but did not always include the methods, or placed them in the wrong class.
b)	Most candidates included two object diagrams for this part and gained full marks. Marks were lost for not explaining why the invalid one was incorrect, or they included diagrams that did not match their class diagram. In some cases, the diagrams produced were not in fact invalid.
c)	A substantial number of candidates did not attempt this question, or included code that was not OCL, such as Java. The candidates that did attempt it, however, often produced answers that obtained full marks.

B4	
	Syllabus area: Section 2 (Concepts) Parts 2.4, 2.2, Section 4 (Practice) Part 4.3
	Total Marks Allocated: 25
	Examiners’ Guidance Notes
a)	43% of candidates attempted this question, but only around one half of the answers produced were good. The question asked for a real-world practical situation in which it would be appropriate to use ad-hoc polymorphism, but many candidates merely showed the syntax for this technique rather than showing that they were able to identify a practical scenario in which this technique would be appropriate to use.
b)	Similarly to question part a), a practical situation was asked for that might lend itself to the use of hierarchical inheritance, along with stub code. Many candidates were able to show the syntax, but fewer were able to identify a realistic scenario in which this type of inheritance would be an appropriate choice. The question also asked for class diagram and stub code, showing the there was some confusion between hierarchical inheritance and other inheritance-based inter-class relationships.

B5	
	Syllabus area: Section 2 (Concepts) Part 2.4, 2.2, Section 2 (Design) Part 3.2, Section 4 (Practice) Part 4.3
	Total Marks Allocated: 25
	Examiners’ Guidance Notes
a)	90% of candidates attempted this question, and a high proportion wrote a good answer. The first part of the question asked students to provide a situation in which it would be appropriate to use method overriding. Since this is a relatively well known technique, most candidates were able to provide a coherent description of where it would be used, with some also showing the syntax needed.
b)	In this part of the question, candidates were asked to discriminate between aggregation and specialisation. The majority of answers achieved this, although again most candidates relied upon being able to demonstrate syntax in a rather application-agnostic way that did not demonstrate a full understanding of where these techniques would be used in practice.

B6	
	Syllabus area: Section 4 (Practice) Part 4.3
	Total Marks Allocated: 25
	Examiners' Guidance Notes
	22% of candidates attempted this question, with about one half of those answers being satisfactory. The question required that candidates implement code suitable for the retention of a stack data structure in an object oriented language of their choice. Marks were lost by neglecting to include all requested operations and test code, or by including functions that alluded to a lack of understanding of the syntax of the language selected or a lack of understanding of what a stack is. Some candidates also chose member visibility settings that were not appropriate (such as designating data as public). Lastly, despite the question clearly stating that none of the methods should display to the screen or take input from the console, quite a large number of candidates did this anyway and so their implementation was not based on the appropriate passing of messages between methods and the test program.