

BCS Higher Education Qualification

Diploma

October 2023

EXAMINERS' REPORT

Object Oriented Programming

Questions Report:

A1	<p>This question was poorly answered and had a poor pass rate.</p> <p>Part a) involved discussing why procedure-oriented programming might be a more prudent choice than object-oriented, but many candidates did not attempt this part.</p> <p>Some candidates gave a lengthy description of different types of languages, but little else. It is acceptable to include some background, but it is important to ensure the answer considered some scenarios to show why procedure-oriented might be a better choice.</p> <p>To gain a high mark, a real-world scenario was needed, which some candidates struggled to do.</p> <p>Part b) asked the candidates to compare and contrast the concept of abstract data types and the class. This part produced better answers, however, some candidates mixed up abstract classes and abstract data types.</p> <p>Some of the better answers included a good comparison of abstract data types and classes but did not always discuss how they differed.</p>
A2	<p>Most candidates answered this question. However, they did not read the question carefully enough.</p> <p>They did not notice that part a) asked about operator overloading, rather than method overloading, which was covered in part b).</p> <p>Most candidates scored good marks in part b), where they were asked to contrast the concept of method overriding with method overloading.</p> <p>To gain a high mark, examples of code fragments were needed to illustrate the points made. Detailed code is not expected, though sufficient code should be included to illustrate the concept.</p> <p>Some candidates lost marks by mixing up overloading and overriding.</p>
A3	<p>A significant number of candidates produced good answers for this question.</p> <p>Both parts a) and part b), looked at different aspects of SOLID.</p> <p>Part a) was about the single responsibility principle and most candidates understood this meant classes should only have one responsibility, rather than several. Candidates lost marks by not considering how this could work in practice, with some thinking this meant a class could only have one method overall.</p> <p>Part b) looked at the open/closed principle in SOLID.</p>

	<p>Most candidates could state that this meant a class should be open for extension but closed for modification. However, they did not always go into further depth, which was needed to gain a high mark.</p> <p>An example code was needed to illustrate how the open/closed principle could be implemented and to gain a high mark, some explanation should be included to show understanding of how the code allows a class to be extended without being modified.</p> <p>Candidates lost marks in this part by not including appropriate examples or including very basic examples of inheritance, without any explanation of how this helps with this principle.</p>
B4	
	<p>Part a) 90% of candidates attempted this question and they answered it very well and achieved high marks.</p> <p>Most candidates were able to attempt a use-case diagram, but some answers lacked details that were explicitly mentioned in the written description, some lacked a system boundary, and some used incorrect symbols e.g., lack of actors.</p> <p>In some instances, candidates did not seem to know what a use-case diagram looks like and drew something else.</p> <p>Part b) Candidates made some reasonable suggestions as to why use case diagrams are used, including that they help to ensure that during development, the system can support the main actors in completing the tasks that they need to be able to undertake.</p> <p>Some good answers also commented that the use case diagram is relatively understandable to a non-technical audience and can be used to support discussions with clients, to ensure that the requirements have been accurately captured.</p> <p>Many also noted that the use case diagram can be used in the development of test cases, to ensure that actors can complete their tasks successfully.</p>
B5	
	<p>Part a) Only a few candidates attempted these questions, and they scored good marks.</p> <p>Most were able to identify which object diagrams were valid/invalid and were able to provide convincing explanations as to why this is the case.</p> <p>In some cases, candidates indicated which they believed to be valid/invalid but did not provide an accompanying explanation, as clearly requested in the question.</p> <p>Part b) Most candidates attempted code for this section, but some failed to spot that the noOfTrains is a class variable (underlined in the class diagram). They did not glean this from the part of the question that suggested that the class variable should be incremented appropriately.</p> <p>Some failed to provide constructors, despite these being clearly requested in the question, providing only the basic class definition. Most, but not all, set the instance variables to an appropriate visibility.</p>
B6	
	<p>Part a) In this question, some candidates did not provide one example from each classification.</p>

	<p>Some provided many examples, with low level detail, from each category. This did not achieve high marks because the descriptions were not accurate.</p> <p>In other cases, candidates described three design patterns, but not one from each classification. In these instances, only marks for one pattern from each group were awarded.</p> <p>The most popular design patterns selected were observer, singleton, and decorator.</p> <p>Part b) There were some good answers to the question of discriminating between black and white-box testing.</p> <p>Some answers were quite vague or conflated these ideas with those of unit testing and integration testing, among others.</p> <p>Some answers also merely described what is meant by these terms but did not address the second part of the question that asked for their advantages and disadvantages.</p> <p>Most stated that black box testing examines the output for a given input to see if it was what was expected, but some descriptions of white box testing were vague, suggesting that the code was tested "line by line".</p> <p>In most cases, candidates pointed out that white box testing is more details and time consuming, and that it requires a greater level of technical expertise.</p>
--	---