Certificate

October 2021

EXAMINERS' REPORT

Software Development

General comments

There were no significant issues with this paper; no particularly unpopular questions and no questions that gave rise to particularly poor answers.

Questions Report:

PART A	
A1	Syllabus 1.1, 1.2, 1.3, 1.4 Overall, this was a very popular question. However, candidates struggled with parts c) and d), and some were missing answers.
a)	Overall, candidates performed well for this part of the question. The subpart regarding the use of *selection* in an algorithm was unfamiliar to some candidates. Otherwise, the best responses contained well informed answers with simple examples.
b)	Most candidates understood the main benefits of pseudocode as one of the main stages in writing code.
c)	There was a range in quality of answers for this part, with a number of candidates struggling with the question. There also appeared to be a significant lack of ability in coding amongst candidates. There were some candidates who could not accurately express the specification in pseudocode.
d)	Again, similar to part c), a wide range of answers were provided. A significant number of candidates had clearly never drawn a flow chart, being unfamiliar with the shapes, such as a decision box. This problem was apparent where students could not express their pseudocode directly as a flow chart.
A2	Syllabus 2.2 3.1a 3.2 This question was fairly popular with well over half of candidates attempting it. However, it appeared that many candidates struggled with this question.
a)	Many candidates lost marks by not providing a trace table of all outputs after each line in the code. A trace table is unambiguous. Simply listing the output of variables when the code had reached the end of execution was a common interpretation of the question and resulted in marks being lost.
b)	A number of candidates failed to justify their choice of data types.
c)	Candidates were able to supply a solution in the programming language of their choice. Many supplied code that would generate the required output, but which was not the most efficient solution. Despite some good attempts, sadly many candidates were unable to express CalculateArea as a function with a return statement, followed by a call to the function from a main program.
A3	Syllabus 2.1 Many candidates produced answers lacking the appropriate depth. However, a small number of candidates achieved full marks on this question. This question was wholly

	descriptive but those candidates who scored low marks would benefit from revising the subject more thoroughly in future.
a)	Functional programming is referenced in section 2.1 in the syllabus. Very few candidates could describe the paradigm to which functional programming belongs. Many of the answers demonstrated that candidates had not encountered a formal definition of this term.
b)	This part was straightforward and required an explanation of modular programming. Most candidates gave good and informative answers, with some using appropriate examples to emphasise the main points. The main requirement was to explain what the process of modular programming is.
c)	Very few candidates could explain their reasons why OOP is either compatible with modular programming or was not. Instead, many candidates spent a lot of effort simply describing OOP in general and not clearly identifying the overall similar approach to software development to modular programming.
A4	Syllabus 4.1, 1.3 3.3 This was the least popular question in Section A, and this was reflected with many candidates struggling to answer the question. All three parts seemed to challenge candidate's knowledge and experience to attract 10 marks in all three parts including coding; file handling; file organisation. All three topics are stated in the syllabus, and it is advised candidates revise all topics on the syllabus.
a)	This part required writing code given a specification. Some candidates misinterpreted the specification, often over complicating the code or occasionally simplifying the code with predefined functions such as sort/max for example. Candidates are advised to read the specification thoroughly, and not to make their own assumptions or simply recall code that was unrelated to the specification.
b)	Overall, candidates struggled with this part. It seemed many candidates were unfamiliar with File I/O routines. The chosen languages could include Java; c;c++; python. The best responses were from candidates who referenced the code in part a) revealing their knowledge of coding file I/O writing and reading to/from files.
c)	A number of candidates had difficulty with providing meaningful answers to this part. Many candidates had difficulty addressing advantages and disadvantages, because they were unfamiliar with indexed sequential file access. A small number of candidates performed well, accepting the use of databases overcomes many of the problems of physical files and indexed sequential files in particular. In discussing disadvantages, candidates needed to consider that databases are more commonly used in modern programming.

Part B	Syllabus topics: 1.1, 2.1
B5	A popular question.
a)	The answers given to this part of the question showed that most students understood the idea of recursion, although some explanations were a little muddled. Very few candidates went on to explain how the set of local variables and parameters used by a recursive function are newly created each time the function calls itself, with the function using a stack for storage of calls. A small number of students explained the need to use an IF statement in the definition of the function to force the function to return without giving a recursive call to itself, avoiding an infinite recursion. The explanations of Iteration were generally clearer. Very few candidates identified the three different types of iteration statement.

b)	Candidates appeared to find this question difficult. Only a minority of candidates were able to provide clear code for the recursive function.
c)	This question was generally well answered with most candidates knowing how to call the function, even if they had difficulty coding it.

B6	Syllabus coverage: 3.4 This was the least popular question in Section B, being attempted by fewer than half of the candidates. Those who attempted the question generally made a good job of it.
a)	This question was generally well-answered. Most candidates were able to explain the difference between Linear Search and a Binary Search.
b)	Some very good answers to this question. Most candidates presented an algorithm for a bubble sort or insertion start – these algorithms were generally correctly presented.
B7	Syllabus coverage 2.2 This question was attempted by almost half of the candidates. There were some very good answers.
a)	Most candidates were able to explain what Object-Oriented programming (OOP) is. However, some answers were quite muddled with unclear explanations.
b)	This question was generally well-answered, although there were quite a few answers where candidates gave dictionary-definitions of the key concepts without relating them to the example provided.
B8	Syllabus coverage: 1.4, 2.3 This question was attempted by over half the candidates.
a)	This question was generally well-answered.
b)	This question was also well-answered, with most candidates presenting a table as expected, along with actual results when the code is tested with an appropriate range of test data. Some candidates also considered the need to check for invalid data, possibly to incorporate validation checks on the input.
c)	This question was less well-answered. The question relates to solving a run time error pointing to problems with the logic of the code.
B9	Syllabus coverage: 2.2 This question was attempted by just over half of candidates.
a)	Most candidates were able to discuss major differences between bespoke and off-the-shelf solutions, mentioning cost, time, and maintenance as key issues.
b)	Most candidates were able to provide a range of advantages and disadvantages to each of the approaches. There were some comprehensive responses to this question.
B10	Syllabus coverage: 2.1, 2.5, 3.2, 4.1 This was one of the most popular questions.
a)	This question gave candidates the opportunity to discuss three of the following.

<u>**Topic**</u>	<u>**Terms**</u>
Procedure Calls	Parameters vs Arguments
Data Processing	Flat File vs Two-Dimensional Array
Program execution	Interpreted vs Compiled code
Testing	White Box vs Black Box

Some candidates discussed all four topics. In these situations' marks were awarded for the best three responses. The answers showed that candidates had a good understanding of testing some of the other answers. Other topics were more confused - in particular, most candidates struggled to distinguish flat files from two D arrays.

B11	Syllabus coverage: 3.1 3.2

	Another popular question, answered by a large number of candidates.
a)	This question was generally well-answered.
b)	Most candidates identified what the algorithm finds. Attempts to go through the logic and conditions expressed in the code were less well-presented.
B12	Syllabus coverage: 5.1 The candidates who attempted this question generally made a good job of it.
a)	This question was generally well-answered.
b)	Some good discussion in responses to this question.