
TECH LEVEL

IT

Computer Programming
Report on the Examination

TVQ01006/TVQ01013/TVQ01015
June 2017

Version: 1.0

Further copies of this Report are available from aqa.org.uk

Copyright © 2017 AQA and its licensors. All rights reserved.

AQA retains the copyright on all its publications. However, registered schools/colleges for AQA are permitted to copy material from this booklet for their own internal use, with the following important exception: AQA cannot give permission to schools/colleges to photocopy any material that is acknowledged to a third party even for internal use within the centre.

General information

There was a significant improvement in the overall standard with very few questions that students did not attempt and no questions which caused universal problems. Questions 4, 5, 6, 12 and 18.2 caused most difficulty. For Question 5, this was disappointing as a similar question had been used in a past paper; Question 6 and Question 12 refer to specific items in the specification. Some flexibility was given in the mark scheme on 18.2 as to the type of error identified (allowing 1 mark), though omitting ‘;’ at the end of JavaScript lines is a stylistic preference and not an error. All code included in exam papers is executed to verify that it works as intended.

The advice on the front of the question paper states: “Use diagrams, where appropriate, to clarify your answers.” Some students gained marks on Question 16 for their diagram when their description was weak: a diagram that ‘describes’ or ‘explains’ (e.g. by way of annotation) will always be considered alongside any written answer.

At this level, students are discouraged from using one word answers unless the question clearly asks for this (e.g. ‘State three languages which...’). In some cases, changing the command word to ‘describe’ has elicited slightly more from weaker students and this has been enough to take them over the pass mark. Where a candidate feels that an answer they have given may be contentious, a short line justifying their choice may be appropriate.

Some students continue to use word processors where the medical need to do so is unclear. In some cases, this has been self-penalising in that, without the writing lines for guidance, a candidate has written too much in some areas and appears to have run out of time, or even missed a question. Where word processors are used, it would be helpful to examiners if the work is 1.5 or double-spaced before printing.

Please can centres check their students have signed the front of the answer booklet.

Questions 1 to 5

Questions 2 and 3 were answered very well with Questions 1, 5 4 progressively less successful. A similar question to Question 5 has appeared in a past paper; many students seemed to have glanced at this as a table of information rather than process the code line by line. For Question 4, ‘FIFO’ is a specific item in the specification, also see [https://en.wikipedia.org/wiki/FIFO_\(computing_and_electronics\)](https://en.wikipedia.org/wiki/FIFO_(computing_and_electronics))

Question 6

This was not answered well, with some students using the context of debugging code rather than how the algorithm works. ‘Backtracking’ is one of the classes of algorithm bullet pointed in the unit content. Please note that ‘sorting’ algorithms, part of the unit introduction (see specification), also form part of the content to be examined.

Question 7

This question was well answered. Where students lost a mark, it was usually because ‘source’ and ‘object’ code were the wrong way around. 96% of students understood ‘mnemonics’ to be a key feature of assembly language, a requirement of AO1.

Question 8, 9, 10

No issues.

Question 11

Although a few students did not attempt this question, most drew a diagram that was a simple representation of the problem statement. Students who used a flowchart were not penalised if they presented a simple representation. Due to varying interpretations of what ‘structure diagrams’ are, questions in this area will provide for this in the mark scheme; the ‘key terms and elements’ listed in the specification are not incorrect, though other approaches may be valid.

Question 12

This was the least well answered question on the paper: the presence of Concurrent Version System (CVS) in the specification appeared to have been overlooked.

Questions 13 &14

No significant issues. If a question asks ‘name 2 languages’, a student who provides more than two should cross out any surplus answers.

Question 15

Unsurprisingly, students found 15.2 much easier than 15.1 although the range of permissible answers for 15.1, and the inclusion of this principle in previous papers, might explain this.

Question 16

Some students addressed iterative design clearly, but did not mention the ‘user interface’ part of the question and were limited to 3 marks. In some cases, the diagram illustrated the cyclical nature of iterative design better than the written answer.

Question 17

Students found the task manageable within the time and space allowed; some used the extension pages at the back to continue the flowchart. Centres should ensure continuation sheets are available, if required. Some students were not sure where to put their rough working for this question; again, continuation sheets could be provided for this purpose and the sheets attached (with working crossed through) to the answer booklet.

Question 18

Most students could select five different aspects and addressed the bullet points appropriately. Some explained lines of code in isolation without addressing the “and program” part of the question, which offered the opportunity to demonstrate more secure technical understanding to those who took it.

Part 18.2: In the code, the lack of an end of statement “;” is not “missing on most lines”, nor is it a “syntax error” as the code provided does compile. Its inclusion or otherwise is optional. See <https://www.codecademy.com/blog/78>