Level 3 Technical Level

IT: PROGRAMMING

Unit 2 Computer programming
F/507/6465
Report on the Examination

TVQ01013, TVQ01015
June 2018

Version: 1.0

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

General

It was notable that some student responses abruptly ended where the question space finished. These responses tended not to appear complete. Students should be reminded that there are additional answer spaces at the back of the student booklet where responses can be continued, if necessary.

Questions 1 to 5

These were generally well answered, but with questions 2 and 4 causing most problems.

Question 6

This question was about high-level and low-level languages.

Both parts of the question were tackled well, with some students apparently benefitting from having studied previous mark schemes. This question drew the best response for a short-answer question.

Question 7

This question was about CLAC-language.

Few students were able to answer this question, perhaps having not studied the topic. All aspects of the unit content will be covered over time, and while students will not be expected to study this area in any depth, they should have simple recognition of what it is.

Question 8

This question was about quad programming.

Students were better at answering the first part than the second. As there are few internet resources available on this subject, the mark scheme was applied flexibly to recognise that the organisation of pair programming may be similar. Students were required to recognise that ‘quad’ is ‘four’.

Question 9

This question was about subroutines and arrays.

There was a mixed response to this question, students were better able to answer 9.1, 9.2 and 9.3, than 9.4. Students did not always understand how return values were handled in subroutines, or how to fix the problem. It is always pleasing to see the different solutions students come up with to fix code and, where valid, these will be credited.

Question 10

This question was about data tables.

Some students did not make the link between the ‘list’ in question 9 and variable types in question 10.1, and they could have used this as an answer. Surprisingly, only around half of the marks were gained on average for this relatively straightforward question, even though the mark scheme for question 10.2 permitted a wide range of contexts to be used.

Question 11

This question was about pointers.

Some students came up with interesting answers to this question, broadening the range of examples covered in the mark scheme. This was a question which students tended to leave blank if they did not know the answer, although some answered that a pointer was a mouse or a stick their teacher used to point at the board.

Question 12

This question was about variables and selection.

This was answered much more favourably than the other coding question (Question 9) though how to correct the code still proved the most challenging part. Some students pointed out that variable n2 also ‘stored the highest number’, for which they were credited, even if that was not the intention of the question.

Question 13

This question was about software design techniques.

Students responded very slightly better to the first part than the second, but answers were sound on the whole. Attention is drawn to the ‘user interface’ part of AO4, in ‘Demonstrate the principles of good program practice and user interface design’ and consideration should be given to how this can be applied to aspects of AO2 and AO3.

Question 14

This question was about software development models.

This was answered at about the same level as question 13. Where there were weaknesses, students tended to repeat comparisons or make similar points without adding value to their answer.

Question 15

This question was about software maintenance.

Again, weaknesses here tended to centre around using similar reasons for each ‘way’ described. Some students did not appear to have planned what they were going to write and rattled off a number irrelevant or unconnected points, instead of focusing carefully on choosing and ‘describing three ways’ as the question asked.

Question 16

This Section B question was about project requirements in software development.

This question was the best answered of the higher-mark questions. Where students lost marks, it was often because they 'listed four ways' rather than 'described two'. Where students described more than two, only their best two were credited.

Question 17

This was expected to be a more challenging question. There was enough context for the student to pause and think about what was needed. Some students displayed poor exam technique by giving answers that appeared rushed at the start before gaining in strength as the students' ideas became clearer to them. Section B questions are intended to have a level of planning, and students would do well to map out what they intend to say before diving into a response. The mark scheme, as with most high mark Section B questions, was a 'levels of response' mark scheme, and it was very difficult for a student to secure a top-band answer ('clear', 'focused') if the early part of their answer amounted to a stream of consciousness.

Question 18

This question was about features and techniques used in computer programming.

This directly referenced AO3 and students were expected to think about how the logic of the game could be solved (for example) through use of selection, arrays and variables. The response was mixed: some were very technical and thought about how the grid (numbers, black squares etc) could be represented and interrogated in arrays. However, many responses were not planned, or recorded in rather general terms how they would set about the task. Some of the student weaknesses with the topic 'arrays', hinted at in questions 9 and 10, were more visible here.

Stronger students used the bullet points carefully and explained some of the specific coding structures that could be used. Students would benefit from practising on programming challenges, which might be solved through coding, pseudocode, flowcharts etc. 'Explain' does not necessarily mean 'write an essay' and in this question short sections of code could be given as a basis to 'explain' how the logic could be implemented, perhaps through annotation, or a list of variables and arrays with an explanation of their utility.

Mark Ranges and Award of Grades

Grade boundaries and cumulative percentage grades are available on the Results Statistics page of the AQA Website.

Converting Marks into UMS marks

Convert raw marks into Uniform Mark Scale (UMS) marks by using the link below.
UMS conversion calculator