Version 1.0 0712

General Certificate of Education (A-level)
June 2012

Computing COMP1

(Specification 2510)

Unit 1: Problem Solving, Programming, Data Representation and Practical Exercise

Report on the Examination

Further copies of this Report on the Examination are available from: aqa.org.uk

Set and published by the Assessment and Qualifications Alliance.

The Assessment and Qualifications Alliance (AQA) is a company limited by guarantee registered in England and Wales (company number 3644723) and a registered charity (registered charity number 1073334).
Registered address: AQA, Devas Street, Manchester M15 6EX.

General

Most students were well prepared for this exam and had made good use of the time available between the release of the Preliminary Material and the day of the exam. There was a higher proportion of students than in previous years who did not attempt all the programming questions in Section D. The questions on this paper were more difficult to predict in advance of the exam and some less able students were clearly unable to develop their own solutions to problems that they had not attempted prior to the exam.

The questions about the Skeleton Program (Section C) were often poorly answered and demonstrated only a limited understanding of the Skeleton Program.

In the *Report on the Examination* for the last few years it was stated that students will not receive marks for screen captures on programming questions where no evidence of actual code has been included and for screen captures that have not been produced by running the student's code. There continues to be a number of students who provide screen captures that have not been produced by the programming code they have given in their answer. (Students often realised parts of their code did not work and removed those parts when completing the tests.) Students who do this obtain marks only for the test results that would have been produced with the programming code they have included (as well as any marks for their programming code).

Some students continue to provide only the lines of programming code they had added/modified, rather than the entire subroutine (as requested in the question). Sometimes this resulted in insufficient evidence that the question had actually been answered, particularly where there was no evidence of successful testing, and this meant that marks were not always awarded.

On the programming questions, minor typographical errors in prompts/messages are not penalized if a question asks for a specific prompt/message to be displayed to the user. However, students will not be awarded the mark if there is a more substantial change made to the prompt/message requested in the question.

Prior to the Exam

Following the release of the Preliminary Materials on 1 March school/colleges were asked, if necessary, to contact the relevant AQA Programmer if they needed to make modifications to the Skeleton Program so that it would work in the programming environment being used at their school/college. There were far fewer school/colleges this year which needed to make modifications to the Skeleton Program.

A copy of the Skeleton Program used by the school/college should be included with the scripts sent to the examiner whether or not the Skeleton Program was modified. Many school/colleges are still not doing this despite this now being the fourth COMP1 exam.

Electronic Answer Document

The Electronic Answer Document (EAD) was made available to school/colleges on 1 March. School/Colleges are encouraged to distribute copies of the EAD to the students so that they can practice using them. A fresh copy of the EAD, not used by any student, must be used in the actual exam.

On some scripts students had taken screen captures of programming code that were sometimes quite difficult to read. It is preferable to copy and paste code into the EAD. This is possible in most of the programming environments used.

Most students knew how to take a screen capture of just the current window, rather than the whole screen, this is something that school/colleges are advised to get students to practice prior to the exam.

Question 1

The majority of students got full marks for this question.

Question 2

Part 2 was generally well-answered. For part 3, some students did not use the number of bits specified in the question and some used even parity instead of odd parity. Part 4 was the first COMP1 question about Hamming code. Many students were able to give an advantage of Hamming code although occasionally answers were too vague, eg, "It can detect errors," and there were some students who clearly had no understanding of the topic and were just guessing eg, "It uses less memory."

Question 3

There was a higher proportion of students this year who included their working for the calculation question (part 5) in the EAD - meaning that they could get a mark for correct working, even if their final answer was incorrect. There is still a significant number of students who are not including their working – this means that if they get the answer wrong they can’t get any marks. Answers for part 6 were often vague and many students provided only a rephrase of the question as their answer. Parts 7 and 8 were the first COMP1 questions about MIDI and this topic was not well understood. More students were able to give an advantage of MIDI than could state an item of data that would be stored about a note. Quite a few students thought that MIDI was used to store samples taken from an analogue sound.

Question 4

Most students did very well on this question with many getting full marks. The most common mistakes were caused by not reading the question carefully and giving either an answer for part 10 that was one of the examples used in the question paper or a permutation that consisted of more than three inputs.

Question 5

Image representation questions have appeared in several previous COMP1 exams and this year’s paper contained a mixture of questions similar to those on previous papers and questions that assessed different aspects of this topic.

The explanation of why more than one bit was needed (part 11) was answered well by many students and the majority were able to work out the correct bit pattern for part 12. For part 13, students who did not provide any working for the file size calculation were unable to get any marks if their final answer was incorrect. Most students were able to give some of the data that would be stored about a vector graphic object, but few got all 3 marks available for this question. Similarly, most students could give one advantage of vector graphics, but few gave two correct advantages. The most common correct answer was that vector graphics do not lose quality when enlarged; it was not enough to say that vector graphics do not pixelate – the concept of “when enlarged” was needed for the mark for this advantage to be credited.

Question 6

For the first time a flowchart was used to represent an algorithm in a COMP1 exam. There was no increase in difficulty resulting from this and the standard of answers was the same as seen in the previous year.

Some students did not follow the algorithm given and instead developed their own program to convert binary to denary. This resulted in them not getting many marks as they had not answered the question.

Students using VB6 tended to get lower marks on this question than those using the other languages available for COMP1. This was partly due to not providing the correct evidence for the testing (screen captures needed to show the data entered for the test as well as the result of the test), although many students using VB6 also seemed to have weaker programming skills.

Students need to be aware that an algorithm is not the same as a program and that simply copying the algorithm into their development environment will not result in a working program in any of the COMP1 programming languages – the pseudo-code/flowchart needs to be adapted to match the syntax of the programming language they are using. As in previous years, a number of students simply copied parts of the algorithm into their program code eg trying to use a keyword of OUTPUT. These appeared to be less able students who generally struggled on the Section D programming as well. The vast majority of students were able to convert the algorithm successfully into working program code and the marks obtained on this question were virtually identical to those achieved on Section B on the 2011 COMP1 exam.

Question 7

Answers to Section C were often of poor quality and very few students achieved good marks on this question.

A number of students are still including additional code when asked for the name of an identifier (parts 22-25). This means that they are not getting the marks for these questions as they have not made it clear which entity is the identifier (sometimes there is more than one identifier in lines of code that they have copied from the Skeleton Program). To reduce the chance of errors, when asked to give the name of an identifier students should be encouraged to copy and paste the identifier from the Skeleton Program, rather than typing the identifier into the EAD.

Very few students showed any understanding of binary files, even though these were used in the Skeleton Program. Part 30 was answered better than most other parts of Section C with most students able to give at least one reason why the use of global variables should be avoided. The majority of students were also able to state an advantage of using a named constant for part 28.

Question 8

This was a fairly straightforward programming question with most students getting close to full marks. Some students did not check their code carefully and subtracted one from `NoOfCellsSouth` or `NoOfCellsEast` (instead of adding one).

Care needs to be taken with screen captures of testing as for part 35 a number of students showed the after state of the cavern and the selection of option D, but did not show the original state of the cavern and thus the screen capture(s) provided did not include sufficient evidence for the mark to be awarded.

A common mistake made by weaker students in all Pascal, VB and Java was to try to combine into one instruction (using a `AND` Boolean operator) an instruction to increment the `NoOfCellsSouth` and an instruction to increment the `NoOfCellsEast` – suggesting that they did not know how to write a case statement that contains more than one instruction.

Question 9

A number of students had clearly anticipated that this question would be asked and prepared thoroughly for it. Weaker students struggled to write the correct conditions for the selection structures and often wrote code that would either prevent all moves in the northernmost row of the cavern or all moves northwards.

A number of answers included code to prevent the player moving out of bounds in each of the four possible directions (and some also prevented illegal moves in a southeast direction as well). This was not necessary as it was not what the question asked. Some weaker students ended up with more errors in their answers by trying to add (incorrect) code to prevent the other possible illegal moves.

Question 10

Most students obtained marks on this question. A number of students did not follow the question specification and changed the messages to be displayed to the user or added one to the `NoOfMoves` variable in the wrong place (often this was done inside the repetition structure used to ensure that a valid move had been entered – this would mean that the `NoOfMoves` variable would be incremented even when a valid move had not been entered). Students should be aware that if a question specifies a particular message to display then this is the message that their program must display – minor typos were ignored, but when a message was different by a whole word or more the mark was not awarded.

Question 11

This was the most challenging of the programming questions and was a good discriminator between students. It was pleasing to see some interesting answers to this question where able students had clearly thought through the problem and come up with their own method for solving it under exam conditions. One unusual correct answer seen from a few students was to pass a copy of the Cavern array to the `CalculateDistance` subroutine and use a loop inside the routine to count how many calls were made to the `MakeMonsterMove` subroutine until the monster and player were in the same cell.

The most commonly used method to calculate the distance was to subtract the monster's east value from the player’s east value followed by a selection structure to deal with the scenario of a negative difference, then to do the same for the difference between the two south values and finally to add the two differences together. A number of students lost marks by dealing with negative values after adding the east difference and south difference together – this would only calculate the correct distance between the monster and player under some circumstances.

It was disappointing that a significant number of students did not include any attempt at answering the question. There was a mark available just for creating a correctly-named subroutine (even if the subroutine did not do anything or use any parameters). Students should be encouraged to include partial solutions to questions they have not been able to answer wholly successfully.

Less able students often struggled to create a new subroutine even though there are numerous examples of subroutines in the Skeleton Program.

A number of students, particularly those using VB, developed a solution that would correctly calculate the distance between the monster and the player but did not set up a mechanism to return the distance to the calling routine. This was often because they had used a procedure, rather than a function (although a few students did use passing by reference correctly as a return mechanism).

Statistical data and information on grade boundary ranges www.aqa.org.uk/over/stat.html

UMS conversion calculator www.aqa.org.uk/umsconversion