AQA

**General Certificate of Education (A-level)
June 2011**

Computing   **COMP1**

(Specification 2510)

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

Report on the Examination

General

Most candidates 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. As was the case last year, most candidates attempted all the programming questions and were able to get some marks on them. The theory topics were, again, not as well answered and responses to the questions about the Skeleton Program (Section C) often demonstrated only a limited understanding.

In the *Examiner's Report* last year it was stated that candidates will not receive marks for screen captures on programming questions where no evidence of actual code has been included or for screen captures that have not been produced by running the candidate's code. There continues to be a number of candidates who provide screen captures that have not been produced by the programming code they have given in their answer (candidates often realized parts of their code did not work and removed those parts when completing the tests). This was particularly true for Section B. Candidates who do this only obtain marks for test results which would have been produced with the programming code they have included (as well as any marks for their programming code).

Some candidates provided 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 by the candidate, particularly where there was no evidence of successful testing. In future years, it is unlikely that candidates will be able to obtain all the marks available for a Section D programming question unless it is clear that the code has been added to the right place in the Skeleton Program. Candidates should also only copy and paste the code for the relevant subroutine(s) into their Electronic Answer Document (EAD); if they include extra programming code then they have not provided the exact evidence asked for in the question and have not shown that they understand what a subroutine is.

Candidates should be encouraged to check their answers to Section D carefully as there were a number of candidates who copied and pasted the incorrect programming code into the EAD, but the screen captures provided as evidence of testing suggested that the correct modifications had been made to the Skeleton Program.

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, candidates will not be awarded the mark if a substantial change is made to the prompt/message requested in the question.

Prior to the Exam

Following the release of the Preliminary Materials on 1 March centres 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 centre. Centres are reminded that the role of the Programmer is to confirm that changes made are acceptable; not to actually suggest the changes that need to be made or to provide advice regarding possible questions on the exam.

A copy of the Skeleton Program used by the centre should be included with the scripts sent to the examiner whether or not the Skeleton Program was modified. Many centres are still not doing this.

The Electronic Answer Document

The EAD was made available to centres on 1 March. As was the case last year, there were clearly some candidates who had not seen the EAD until the day of the exam. Centres 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 candidate, must be used in the actual exam.

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

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

Question 1

Most candidates obtained the mark for part 1. Those who did not get the mark often used 7 bits instead of 8 – 1111011 - (missing out the 0 on the LHS) or miscounted the number of 1s to use – 0111011. Part 2 was generally well-answered. The most common incorrect answers were 255 (the highest decimal value that can be represented using 8 bits) and 128 (the number of different values that can be represented using 7-bit binary). This suggests that candidates are recognising that they need to do 2^8, but often then get confused about when they need to take a 1 away – getting muddled finding the highest value with finding out the number of values that can be represented. Some candidates are taking 1 away from 2^8 and some are taking 1 away from 8 and giving an answer of 2^7. A larger number of candidates got the correct answer to the hexadecimal question this year, but a significant proportion did not understand what hexadecimal is and made either no attempt or gave an answer that was not in hexadecimal. Answers for part 4 often lacked precision and did not convey the idea that it is easier for a person to read the hexadecimal equivalent of a binary bit pattern. A common misconception is that hexadecimal values will use less storage space.

Question 2

This was the first question that had been asked about Gray codes and it was clear that most candidates have had little exposure to this topic. Few candidates were able to get all the marks on parts 5-7. For part 8, the most common correct answer was that Gray code counters use less electrical power. A number of candidates simply stated that only 1 bit changes at a time; this is true but does not answer the question which was to give an advantage of Gray code counters. A common incorrect answer was to state that Gray code counters use fewer bits (they would use the same number of bits), that they are more efficient (not a precise enough answer) and that they are easier to read.

Question 3

Many candidates could give the correct definition of the resolution of an image for part 9. The most common mistake was to give a definition for the resolution of a VDU instead of the resolution of an image. Fewer candidates were able to define colour depth. The most common wrong answer was to state that the colour depth was the number of colours, rather than the number of bits used to represent the colour of a pixel. Most candidates were unable to calculate the file size for part 11. Candidates should be encouraged to show their working out as this may allow them to get some marks even if their final answer is incorrect. Most candidates were able to give an advantage of vector graphics, although some answers were too vague to be creditworthy. Good answers made it clear that the quality of a bitmapped image deteriorates as it is enlarged whereas a vector graphic does not.

Question 4

A straightforward question, but many candidates had clearly not prepared for this topic and either did not attempt the question or were clearly guessing. It is important that candidates are aware that all topics on the specification can be assessed – even smaller topics. More candidates identified Testing than Design as one of the missing stages in the systems development life cycle.

Question 5

This was the first COMP1 question in which candidates were asked about the theory of problem solving. Most candidates received good marks and were able to identify the goal and three resources that were available. Far fewer candidates were able to define clearly what is meant by the ownership of a problem with many stating that the owner was the person affected by the problem, rather than the person responsible for solving the problem. Another frequently seen incorrect response was to write who had ownership of the problem described in the question (Bob), rather than to define ownership.

Question 6

Many candidates were able to identify that the algorithm in Figure 3 was a bubble sort; a significant number of these were also able to complete Table 2 correctly. Some candidates worked out (or guessed) that the algorithm was a bubble sort and then completed Table 2 by filling out all the Scores in the correct order instead of doing what the question asked which was to complete one pass through the outer loop. Candidates must ensure that they read questions carefully and answer what has been asked.

Question 7

The Section B task (programming unrelated to the Skeleton Program) was a more challenging question than those on the 2009 and 2010 COMP1 question papers. However, it was based on a standard algorithm (linear search) that is on the specification. Despite the Preliminary Material clearly stating that candidates should be familiar with declaring and using arrays (and there being examples of arrays in the Skeleton Program), a significant number of candidates were unable to write a syntactically correct array declaration in their programming language. A number of candidates provided screen captures that had not been produced by the programming code they had given in their answer for part 20; this meant that they did not get any marks for their screen captures. Candidates should understand that they could get marks for test runs which show only part of their program working correctly, but they will not get any marks for "correct" test evidence that was not produced by their programming code.

Most candidates were still able to score good marks on this question despite the increased difficulty of this task.

Question 8

Most candidates were not well prepared for this section and did not do as well on these questions about the Skeleton Program as they did on the questions where they were asked to modify the Skeleton Program. In particular, little understanding of structure charts or decision tables was shown by a significant number of candidates.

It was pleasing to note that most candidates only gave the name of an identifier when asked to do so – those who copied and pasted sections of code from the Skeleton Program did not get the marks for these questions as they had not demonstrated that they understood what an identifier is (some candidates gave answers that contained multiple identifiers). Some candidates did not get the mark for giving an example of a constant declaration as they provided only the name of the constant. Candidates should ensure that when asked for the name of an identifier they provide only the identifier in their answer and when asked for an example of a type of program statement that the entire program statement is given in their answer.

For part 36 many candidates described the repetition structure rather than the selection structure inside the repetition structure.

Question 9

Most candidates attempted this question and were able to get the majority of the marks. Despite the question asking that the new option of "RUN OUT" be available for both real and virtual dice versions of the game, a number of candidates did not alter the Skeleton Program to generate a random number between 1 and 5.

Question 10

For question 10 candidates were asked to adapt the `DisplayResult` subroutine so that an appropriate message would be displayed if the result of a game was a draw. Many candidates got good marks on this question. The most common mistake was to add an else clause to one of the existing `IF` statements rather than adding an additional `IF` statement – this would result in the message about a drawn game being displayed if one of the player's had won the game as well as when a game was drawn. Some candidates adapted the Skeleton Program correctly, but then did not provide evidence for the test asked for in the question – a test showing both players getting a score of 0 was needed. Some candidates provided test evidence when the players have obtained a score of 1 or more.

Question 11

While there were a lot of good answers to this question, candidates generally found question 11 more difficult than questions 9 and 10. Candidates often used the incorrect logic. Common mistakes included using the wrong logical connective for the two conditions (i.e. AND instead of OR / OR instead of AND) and using the wrong logical operator with a numeric value e.g. ">=6" instead of ">6" or ">=7". It was clear that a significant proportion of candidates following the AS Computing course struggle to understand the logic of selection/repetition structures which have multiple conditions. A number of candidates did not read the question sufficiently carefully and did not include a repetition structure inside the `RollBowlDie` routine – only using a selection statement.

Question 12

Many candidates had clearly anticipated that they would be asked to write a routine to save the top scores to a file and did very well on this question with able candidates often obtaining full marks. Some candidates seemed to have tried to memorise the code for this task and then were unable to reproduce it under exam conditions (or simply copied and pasted the `SaveTopScores` subroutine and then tried to modify it) as they did not sufficiently understand the task they had been practising. For part 47, a number of candidates did not modify the main program block to allow the 5th option to be selected.

Question 13

A wide range of responses were seen to this question. A large number of candidates were unable to express their ideas clearly and their description of how their suggested changes could be made was too vague to get full marks. Some answers would have achieved the desired result of getting the low scores more than the high scores, but also resulted in adverse, undesired changes to the Skeleton Program (e.g. a player could no longer get 2 runs and could never get a result of "out").

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

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