AQA

**General Certificate of Education (A-level) Applied
January 2012**

**Applied Information and
Communication Technology**

IT09

**(Specification
8751/8753/8756/8757/8759)**

Unit 9: Software Development

Report on the Examination

Unit 9: Software Development (IT09)

The format of the examination is an AQA-set assignment, for which students are allowed time for research and design work (the Investigation Time), then a period of Controlled Conditions during which students are expected to produce their software system and an evaluation of the product and their own performance.

Commentaries on exemplar work from previous series' may be accessed through Secure Key Materials.

General Comments

Most students attempted to produce software systems that addressed the task set, thus being able to access the full range of marks. Some scored high marks.

The Task

The task given for this examination series was:

'A small, independent dealer sells single items and groups of items. She has a national and international trade.

She wants you to create a software system that will allow her to add, amend and delete customer records and to record, search for and amend orders that come in. She also wants to be able to perform statistical analysis on her sales and her buyers.'

Items (a) to (g) of the task are produced during the Investigation Time, whilst items (h) to (m) are produced during the Controlled Conditions.

Items (a) to (g) produced during Investigation Time

Most students correctly interpreted the task and produced systems that allowed the trader to perform the required tasks.

Item (a)
Those students who provided a task based time plan and time estimates for each task were awarded two marks.

Item (b)
Most students described a relevant client and why a software system was required. Students who provided a reasonable description of why the system was needed were awarded two marks.

Item (c)
Most students were awarded one mark for identifying an intended user or users and their relevant IT skills, or lack of them. Better students were able to clearly link the skills and experience of the users to elements of the proposed system's user interface and were awarded two marks.

Item (d)
Most students stated what inputs and outputs were required to achieve the task set. However some did not provide detailed specifications of either. Inputs described often included clicking buttons but neglected data inputs to data storage structures. Descriptions of outputs were less well described than inputs and it was rare that on-screen reports or printed reports were described in detail – often just the general name of the output was given.

Item (e)
Most students produced evaluation criteria that included qualitative and quantitative criteria. Many students incorrectly identified the types of evaluation criteria, often identifying qualitative criteria as quantitative and vice versa.

Item (f)
Few students specified the folder structure to be used, though many did provide screenshots to show how their files were stored. In addition, some students showed that they understood why this was necessary and why it was important to save versions of their software system as it was being developed.

Most students made a good attempt at producing designs for the interface of their system, showing the key features. Few explained how the design choices made related to the user needs, though better students did clearly note how the features would meet the user needs listed.

Most students provided some kind of data dictionary that defined the data structures necessary for their system. The majority of these were only sufficiently detailed to be awarded one mark. Better students described typical items of data, validation rules and temporary storage variables and arrays that would be necessary to allow a third party to implement the system.

Item (g)
Some students demonstrated that they understood the different types of tests necessary to test their software system effectively and explained how they would apply them. These students were awarded two marks for their test strategies. Many others appeared confused by what constitutes a test strategy, particularly with regard to testing the whole of the system.

Many students neglected to specify all three types of data necessary for good functional testing: normal data (that is acceptable), extreme data (that will test the boundaries of what is acceptable); erroneous data (that is totally unacceptable). Few test plans included data that effectively tested the whole of the system

Items (h) to (m) produced during Controlled Conditions
Students may only take printed or hand-written material in to Controlled Conditions, and additional material may not be brought in after the start of the first session of Controlled Conditions. Implementation of the planned software system must only be attempted under Controlled Conditions.

All students in this examination series used Microsoft Access and Visual Basic to produce their software system. Most students provided evidence of using programming techniques to produce their system, with some good examples being seen.

Item (h)
Most students provided some evidence of following their test plans but few noted what changes were needed to the solution. Those who did were awarded three marks.

It was disappointing that some students provided virtually no evidence of having carried out any testing.

Item (i)
Some students had appropriately used some student defined program control structures but few had talked about how or why they were being used, which is necessary to achieve the third mark. On the whole the program control structures used by students were restricted to simple, or nested, if statements, though the scope of the system required would have allowed other selection and repetition structures to be used.

Few students had used appropriate student defined variable, object and procedure names, and of those even fewer had actually indicated this through annotation or commented code.

Very few students had identified where modular programming techniques had been used.

Where students had used program variables, or specified data types, few had explained their choices.

Most students provided some annotated evidence of the key features of their software system and in a few cases this was good enough for three marks to be awarded.

Item (j)
Some students had produced instructions for installing the software system on the client's machine and instructions for accessing the key features of the software system. This was sufficient for two marks to be awarded

Some students provided step-by-step instructions on how to create the system, which could not be awarded any marks.

Item (k)
In this item the student's evaluation of their software system and the quality of written communication is assessed.

Better students used well written text to analyse the success of their software systems by comparing the results of testing with the evaluation criteria and the client needs. They used an appropriate form of presentation, often text combined with tabular information. They also used appropriate vocabulary to explain some of the complex ideas in their analysis. On the whole their work in this section was well structured and coherent. It was interesting to note that some of the highest marks went to work which was very concise.

Weaker students tended to neglect the results of testing and often used poorly written English that was not in an appropriate format.

Item (l)
Students who had achieved full marks for item (a) often achieved full marks for this item. Weaker students tended not to explain alterations to their schedule in enough detail to achieve two marks.

Item (m)
Many students discussed their own performance in enough detail to achieve two marks, often discussing lessons learned and possible alternative approaches. Most students did achieve at least one mark.