



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

**Applied Information and  
Communication Technology**

**IT09**

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

**Unit 9: Software Development**

***Report on the Examination***

---

Further copies of this Report on **the Examination** are available from: [aqa.org.uk](http://aqa.org.uk)

Copyright © 2011 AQA and its licensors. All rights reserved.

**Copyright**

AQA retains the copyright on all its publications. However, registered centres 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 centres to photocopy any material that is acknowledged to a third party even for internal use within the centre.

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.

## **Unit 9: Software Development (IT09)**

The format of the examination is an AQA-set assignment, for which candidates are allowed time for research and design work (the Investigation Time), then a period of Controlled Conditions during which candidates 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 the Secure Key Materials site via e-AQA.

### **General Comments**

Most candidates attempted to produce software systems that addressed the task set, thus being able to access the full range of marks. Some scored high marks. Some candidates misinterpreted the task and were not able to access the full range of marks.

### **The Task**

The task given for this examination series was:

“For this assignment, you will need to create a software system to meet the needs of a specified client.

A local estate agency has a software system that allows potential purchasers to search for properties that meet their requirements.

They now need a system that will allow purchasers to enter answers to a series of questions in order to establish their requirements. This system must allow purchasers to save answers that they entered so that they can retrieve them and change their requirements if necessary.

The software system should be designed for a clearly specified client, and meet the requirements of that client. It should also take into account the ICT skills of the intended user(s).”

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 (f) produced during Investigation Time**

Many candidates misinterpreted the task and produced systems that allowed the estate agency to enter records for properties that they were selling or letting. This was not a requirement of the task. Most candidates failed to realise that the users of the system would be potential purchasers as well as possibly the agents themselves.

#### **Item (a)**

Those candidates who provided a task based time plan, listed in a logical order, and time estimates for each task were awarded two marks.

#### **Item (b)**

Most candidates described an estate agency or letting agency and a person within the organisation as their client. Many, however, did not describe why the software system in the task was required. Candidates who provided a reasonable description of why the system was needed were awarded two marks.

#### **Item (c)**

Most candidates were awarded one mark for identifying an intended user or users and their relevant IT skills, or lack of them. Better candidates 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.

It was clear that many candidates had not realised that the intended users of the system were potential house purchasers rather than the estate agents themselves. Candidates who had realised

this were able to talk about general features of the interface design that would meet the needs of this very wide group of users.

### **Item (d)**

Many candidates neglected to reflect the client's stated needs, or to interpret them in the light of the task set. These candidates were not able to achieve all the marks available.

Many candidates stated what inputs and outputs were required to achieve the task set. Few, though, gave 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.

Where processing was described using dataflow diagrams (DFDs) these were often not broken down to a level where the detail of processing could be seen. DFDs that show this level of detail will often be at Level 2 or Level 3.

### **Item (e)**

Several candidates confused the evaluation criteria with the description of the client needs (item (d)) and they were therefore unable to gain maximum marks overall on the two items.

Most candidates produced evaluation criteria that included qualitative and quantitative measures. Many candidates incorrectly identified the types of evaluation criteria, often identifying qualitative criteria as quantitative and vice versa.

### **Item (f)**

Few candidates specified the folder structure to be used, though many did provide screenshots to show how their files were stored. In addition, some candidates 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 candidates 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 candidates did clearly note how the features would meet the user needs listed.

Few candidates showed how modular programming techniques would be used.

Most candidates 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 candidates 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 candidates demonstrated that they understood the different types of tests necessary to test their software system effectively and explained how they would apply them. These candidates 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.

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

### **Items (h) to (m) produced during Controlled Conditions**

Candidates 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.

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

### **Item (h)**

Most candidates 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 few candidates tried to test the whole system by inputting purchaser requirements, saving these requirements and then retrieving and amending them.

### **Item (i)**

Some candidates had appropriately used some program control structures that they had defined themselves 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 candidates 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 candidates had used appropriate candidate defined variable, object and procedure names, and of those even fewer had actually indicated this through annotation or commented code.

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

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

Most candidates annotated some 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 candidates 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 candidates provided step-by-step instructions on how to create the system, which could not be awarded any marks.

### **Item (k)**

In this item the candidate's evaluation of their software system and the quality of written communication is assessed.

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

### **Item (l)**

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

### **Item (m)**

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

### **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 or scaled marks into marks on the Uniform Mark Scale (UMS) by using the link below.

**UMS conversion calculator** [www.aqa.org.uk/umsconversion](http://www.aqa.org.uk/umsconversion).