



**General Certificate of Education (A-level) Applied  
January 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 Subject Manager at 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 very high 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 leisure activity club loans equipment to its members. The club wants to develop a software system for recording the loans. The software system must also allow for a specific record to be found and amended when loaned equipment is returned.

The software system must include facilities to allow the data to be analysed so that the club can, for example, highlight the items of equipment that have been loaned the most in order to allocate spending on new equipment.

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 (g) produced during Investigation Time

Many candidates did not appear to spend enough time thinking about who they would be creating the system for. This was shown in the generally poor marks awarded for items (b) to (e). This poor definition of the client and their needs had a detrimental effect on performance in later items in the task.

#### 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 did describe some kind of leisure club or similar relevant organisation and a person within the organisation as their client. Some, however, did not describe why the software system in the task was required. This was normally because the organisation chosen was not very suitable for the task. 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 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. Some candidates suggested that because the user did not know how to use Microsoft Access the user interface would need to be made simple. This is not a logical conclusion to draw and similar comments were not credited.

**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 of the marks available.

Most 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.

Some good examples of descriptions of the processing were seen, including some with easily understandable pseudocode. Few candidates explained how the processing met the client needs.

It was interesting to note that candidates who had provided very clear descriptions of the processing were able, during the implementation, to produce a system that clearly met the client needs. Candidates whose descriptions were less clear appeared to struggle with the implementation.

**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)**

Most candidates specified the folder structure to be used and showed how their files were consistently and sensibly named. In addition, a large number of 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.

Some candidates showed that modular programming techniques would be used by describing reusable or public modules of code that they would write, or by showing how their form-based system would be linked together. Unfortunately, some candidates provided JSP diagrams that showed the logic of processing rather than the modularity of their designs.

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)**

A good number of 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.

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 good evidence of following their test plans and noted what changes were needed to the solution, or the changes made, and were awarded three marks.

A pleasing number of candidates tried to test the system, but few followed the process through by inputting data, showing the results in the affected modules, amending data and showing the new results. For example, candidates showed that a new loan could be recorded and an output produced but did not then show what happened to the output when the loan was amended.

### **Item (i)**

Most candidates had appropriately used some candidate defined program control structures but few had talked about how or why they were being used, which is necessary to achieve the third mark.

Most had also used appropriate candidate defined variable, object and procedure names, but few had actually indicated this through annotation or commented code.

Few had identified where modular programming techniques had been used.

It was clear, in the majority of scripts, that appropriate data types (both in the data structures and in the program variables) had been used. Few candidates explained the choice of these data types, particularly in the program variables) in order to achieve the third mark available.

Most candidates produced good annotated evidence of the key features of their software system and in many cases this was good enough for three marks to be awarded.

### **Item (j)**

Approximately half of the 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.