Version 1.0: 0710

ASSESSMENT and QUALIFICATIONS ALLIANCE

General Certificate in Education

Computing 2510

COMP4 Computing Practical Project

Report on the Examination

2010 examination – June series

General

Centres that entered candidates in 2010 need to read this report in conjunction with the specific feedback sent to them on the publication of results. The comments below highlight the observations of moderators during this year’s examination and should be used together with the subject specification and assessment criteria and the COMP4: Advice and Information Booklet for the Teachers' Standardising Meetings to ensure that centre assessment is accurate.

Compared with the legacy specification, there were changes in the marks awarded for some sections and the total mark; a new section on data volumes and the requirement to write a User Manual dealing with the functionality of the **whole** system have been added. However, the most important change was the need to correctly identify the complexity level of the problem and its solution from the outset at the task setting stage.

This is the first year that the COMP4 specification has operated. Most centres had taken on the requirements of the new specification and managed to get their candidates to program a solution at the level of which they were capable. However, there were some cases where much time had been spent on coding because of the complexity of the problem and the Analysis and Design sections were probably written post-implementation when time was running out. The User Manual and Appraisal sections were also often casualties of poor time management by the candidate.

In order to avoid possible plagiarism, Centres should **not** encourage their entire cohort to produce CAL systems with just differences in subject area; a variety of projects is expected from individual centres. Some ingenious non-data processing projects were seen this year including one involving iphone programming. Data processing projects are still welcome and often make it easier for candidates to find genuine end users.

Internal standardisation was generally well carried out, but it is probably not a good idea for local assessors to mark only individual sections for their centre without reading through the whole report for each and every candidate.

The better projects seemed to be the ones where candidates had fully coded the solutions, but very good solutions were also seen with genuine client-server applications. Most centres have moved away this year from customised Microsoft Access projects with minimal coding as was intended in this new specification.

Administration

Centres that accurately complete the Candidate Record Form, Project Log Sheet and Centre Declaration Sheet can really assist in the moderation process. Moderators always seek to agree with the local assessor; this is very difficult to do if all that is provided are the section marks and final total on the Candidate Mark Form and perhaps a highlighted perceived level of complexity. The centres that have devised their own form of comment sheets with extensive word-processed comments are also to be commended. However these still need to be accompanied by the Candidate Record Forms. It is still a *mandatory*, QCDA / OFQUAL requirement that the signatures on the Candidate Record Form are completed to minimise plagiarism. These Candidate Record Forms *must* be the current versions obtainable from the AQA website at http://www.aqa.org.uk/admin/p_course.php . Many centres used the Project Log Sheets very effectively to explain their rationale for how they had awarded their marks. It is also useful for the candidates and the teachers to complete the Page Numbers boxes. There were also some very effective examples of script annotation by some local assessors.

There is ample space on the Candidate Record form for the local assessor to justify briefly their perceived level of complexity that they have highlighted for the four categories. This is expected here if not clearly explained elsewhere on centre designed documentation.

Candidates still need to be aware that legacy examination CPT3 contexts must *not* be used for A2 projects because AQA has already completed the analysis for CPT 3.

Too many centres sent in work late, without prior agreement by AQA. This strategy risks the late publication of their own candidates' results that may have serious consequences for them during the UCAS Clearing process. Centres are reminded that it is *their* responsibility to get the projects to the moderators by the deadline date using *first class post*. Moderators spent much unproductive time this year chasing individual centres for coursework. Please do not send projects using a method that requires a signature because most moderators teach full time and cannot be expected to collect work from the courier at their distribution centres. Some of these centres are a long way from the moderator's home addresses and many Parcel Force offices also have very restricted opening hours. Please ensure that sufficient postage is attached.

A number of centres failed to send in Centre Declaration Forms or did not have them countersigned by the Head of Centre. Some centres also had incorrect additions on their Candidate Mark Forms

Projects need to be bound as a booklet ideally with one or two treasury tags to enable them to be easily read. Some projects arrived untagged (loose!) in document folders and a few others in very large ring binders.

Software

The majority of candidates used Delphi or a version of Visual Basic/VB.Net. Other languages seen included VBA, C # or C++, Java, Python, Dark BASIC, ASP, MySQL, PHP, HTML and Actionscript 3. HTML and CSS may have formed part of the solutions presented, but needed to be combined with one or other of the other programming languages. Candidates completing a project entirely with a high level language accounted for most of the very high scoring projects, but a number of very good client-server systems were also seen involving PHP and MySQL for example.

Project Development

Low scoring candidates still tended to submit simple data processing projects with little complexity, but most of these were assessed appropriately by the centres. A small minority still produced customised database applications with minimal candidate code; these were heavily penalised during the moderation process.

There continues to be too many candidates who pursue a project that merely used either a fully coded or a part package, part coded solution to enter data, save it, update it and to produce simple reports with little or no data transformation. Regardless of the programming language used even if it was a fully coded solution, this type of project is not complex enough for A2 and cannot score a good mark however well the report is written and however hard the candidate worked to produce it.

Project Reports

The majority of the high scoring candidates used a programming language or a combination of a package and substantial programming to demonstrate good coding skills and produced welltested, well-documented effective solutions to real problems, with corroborated end user feedback. Candidates who followed the reporting style of the specification and followed this up by completing the log sheets fully usually justified their assessment. More centres used the Candidate Record Forms and Project Log Sheets to record a quantitative indication of the candidates' achievements, rather than to record how much effort a candidate has put into their work.

Most candidates achieving a high mark addressed fully all the items listed in the specification *in the context of their projects*. The project report is not the place for general theory. As in previous years, those who scored the highest marks included well reasoned and justified explanations of all aspects that are listed as indicators in each section of the specification and, in particular, good programming techniques which, together with a high performance level in the other sections, tended to be the feature that most distinguished the high scoring candidates from the others. It would assist the moderators greatly if items are addressed by candidates in the same order as the specification and with the same headings, especially when essential items appear in one or more appendices without accurate references as to where these items can be found. Sampling as described in the subject specification reduces bulk. One candidate produced a testing section in excess of 12 cm in thickness of A4 paper.

Analysis

Possibly, there were fewer thorough analyses than in previous years. It cannot be over emphasised how important this section is to confirm the complexity of the project. A number of candidates provide a structured investigation using one or more formal techniques including interviews, questionnaires, Observation and the use of analyzed source documents. It is not appropriate for the candidate to assume that their personal view of a problem provides enough detail. This happened in the case of a number of Complex projects scoring high Technical Solution marks. What is required is a carefully composed interview, questionnaires where appropriate and evidence of the current system. The best examples of good practice contained a comprehensive list of highly detailed specific objectives constructed from more than one interview with the end user and improved/clarified/checked for completeness after an initial prototyping stage involving feedback from the end user. The finally agreed set of specific objectives were signed off and dated by the genuine end user. The best examples also contained an outline of the proposed solution by dataflow diagrams and a careful consideration of the proposed and alternative solutions. The latter is an aid to confirming the complexity level of the problem.

The analysis data dictionary still remains a problem area with many centres ignoring it or just copying the one presented in Design. The analysis data dictionary captures the data items and their properties that appear on the artefacts of the current system, e.g. a paper-based booking form. The properties will include such items as length/range/size of fields, data types of fields, example values. The data types will be as perceived by the end user, e.g. a whole number, a decimal number with two decimal places, not the data types that will be used in the chosen programming language, e.g. Long Integer, Float(5,2).

Numerous candidates totally ignored the new section on data volumes even though it was relevant in their case because they were tackling a data processing problem. In some cases, the data volumes are relevant in determining the level of Complexity of the project. If it is not relevant for a particular problem, candidates are expected to briefly justify this reason for its exclusion.

Some moderators reported that more diagrams are now being used, but not all candidates produced DFDs for the existing system. Candidates should be encouraged to use standard notation and labelling especially for DFDs and ERDs.

Many candidates produced clear and measurable objectives that were specific to their projects and not generalised, 'text book' type objectives such as 'user friendly' or 'system must load in seconds'. 'SMART' objectives as defined in the current specification enable candidates to achieve higher marks in both this and the Appraisal section. It is good practice to break the objectives down into those that are qualitative and those that are quantitative. The objectives of some candidates often indicated little processing, but centres still claimed these to be 'Complex'. The complexity of a problem can be assessed by examining the specific objectives, the dataflow diagrams of the proposed system, the data model where the latter is appropriate, an outline of the proposed solution and by discussion with the candidate. This becomes more difficult if the candidate fails to provide sufficient detail. In this case, examination of the design and implementation should throw light on the complexity level.

Many identified end users (client-users) seemed to have very little influence on the specific objectives and could provide little or no meaningful feedback in the later Appraisal section. A few candidates got their end user to sign off and date the agreed objectives. This is considered to be good practice.

Please note that the awardable mark for this section is now linked to the level of complexity of a project. If a centre determines the level of complexity as anything other than 'Complex', they cannot then award marks in Band 4.

Design

Some designs were well planned and showed a high level of understanding of the requirements of an A2 project. Many candidates failed to score high marks by not conveying an understanding of the concepts of validation and security beyond a very superficial level, such as field length checks. The high scoring candidates show a good appreciation of data structures, file design and normalisation where this is required for their solution. The normalisation process can be demonstrated by showing how the data model develops from the analysis data dictionary to 3rd normal form using standard notation or by confirming that the entity descriptions derived from the E-R diagram are in 3rd normal form or BCNF using one of the tests, e.g. every non-key attribute is a fact about the key, the whole key and nothing but the key. Some candidates are still failing to show proof of normalisation and just producing ERDs, often incompletely labelled.

There seems to be a great reluctance to show the HCI design as clearly annotated sketches or by the use of a graphic package or the form designer related to the chosen programming language. This allows the candidate to give clear explanations as to the rationale behind the input and output screens, and how they meet the user needs. Chapter 7.2 and 7.6 of the AQA endorsed A2 text book[1] contain guidance on HCI. The HCI design should also convey guidance in pictorial, annotation and prose form that will enable the HCI design to be implemented. For example, the rationale may cover the reason for choice of fonts, font sizes and colours for text, foreground and background as well as specifying these.

[1] 'AQA Computing A2' by Kevin Bond and Sylvia Langfield, published by Nelson Thornes ISBN 978-0-7487-8296-3

A number of candidates appeared to produce extensive post implementation designs without even claiming these to be ‘prototyping’. Such ‘prototyping’ should have corroborated end user involvement by signing off and dating.

Very few candidates gave clear algorithms relating to the data transformation to be programmed by the candidate. These algorithms should be in a form that could be coded in any language and need to concentrate on the complex parts of the system that the candidate is going to code. We are not very interested in the algorithms required for login, form navigation and password changes etc. Algorithms in pseudocode or structured English must be given if a high mark is to be scored. Those that were given were generally poor with the notable exception of some centres whose methodical approach made moderation of this section relatively simple. Frequently ‘algorithms’ were post implementation code extracted from the full code listing, not in pseudocode or structured English as required. The algorithms presented in this section are one of the places that moderators inspect closely to confirm the potential level of complexity of the project.

If candidates are going to need to embed SQL queries in their programming code because they will be creating part programmed, part package solutions, it is appropriate to include these query designs in this section. Similarly, DDL script designs for table creation should be included here as appropriate.

Security, integrity and test strategy are problem areas as many candidates are still not addressing these in the context of their system, but presenting general theory rather than applying it to their particular projects.

Technical Solution

This section is now also closely linked with the level of Complexity. Many centres had managed to get their candidates to program a solution at the level at which they were capable , but there was a great deal of evidence of much time being spent on coding and when this was completed or time was running out, the Analysis and Design sections were attempted. The complexity present in the technical solution is one way that the complexity of the problem as defined at analysis can be confirmed. A technical solution that has some embedded SQL or that creates objects at runtime, for example, doesn’t in itself make the problem being solved complex. However, if at the Analysis phase doubt is present as to the appropriate level of complexity then evidence from Design and Technical solution phases should be used.

Only some local assessors made clear statements on the Project Log Sheets or other centre generated documentation such as “A fully working robust system”, “most processing objectives achieved” etc. Those that did, significantly assisted the moderation process.

Many of the candidates who used a programming language produced well-structured listings. High scoring candidates used advanced features appropriately. It is helpful if such code actually written by the candidate is highlighted by the local assessor and, in particular, the use of advanced features such as user defined data/record structures need to be clearly indicated. This confirms that the candidate understands what they have done. There is no longer a specific reference to parameter passing in the specification; this is assumed as demonstrating high levels of technical competence where parameter passing is appropriate. Solutions are not ‘Complex’ simply because parameter passing has been used.

There was a surprising mixture of candidates with a high mark in this section gaining a high mark overall and those who seemed to spend the majority of their time on their coding to the neglect of the other sections that account for 55 of the 75 marks.

If a part programmed, part package approach has been adopted, it is important that validation is coded by the candidate using the programming language and that SQL queries are embedded within the code as discussed in the 2010 COMP4 Teachers' Standardising Meetings. Similarly, the reports also need to be programmed. A small minority of candidates again only customised a database package with minimal amount of self-written code. This is not in keeping with the philosophy of the new specification. Producing a GCSE level football league table in a spreadsheet package with a few cell equations and no coding is similarly inadequate at A2 level.

The Computing specification does *not* require an ICT-style implementation guide. Evidence for the mark in this section comes from the testing and system maintenance sections and possible the user manual where screen shots may be the only evidence of the working system if other sections are not complete.

System Testing

Testing is one of the few sections that is assessed without reference to the perceived level of complexity. Testing by many of the average and weak candidates was trivial. It is *not* appropriate to have multiple tests for login passwords and/or to prove that all the buttons navigate to other forms and / or print reports. It needs to be emphasised that what is required is proof that the candidates' coding works to produce accurate results. The higher scoring candidates showed evidence of carrying out a thorough test plan effectively by producing a table of expected results and screen shots to prove it; the screen shots were well annotated, *not* heavily cropped or just labelled a figure number and were cross referenced to the table entries and ideally corroborated by the assessor or end user.

Few candidates used boundary testing effectively. In some cases boundary data was non-existent because of the type of problem being solved, but this needs to be clearly stated and justified. Sampling is needed for this section to avoid coursework of massive thickness as mentioned earlier.

Maintenance

Many centres used Prettyprinter software to produce the code listings and most candidates had used appropriate, self-documenting variable names. Many candidates also annotated their code appropriately. Frequently the procedure and variable listings (including scope as appropriate) were absent. This would make maintenance of the system difficult. Candidates who had programmed in Java often had appropriately used the Javadoc tool to create some of the documentation for this section.

Not all candidates produced algorithms or a suitable alternative and only a small number of high scoring candidates produced them at a level that would enable *their* systems to be effectively maintained. If the algorithms originally proposed in Design have been changed during Implementation then they need to be updated here. If they have not been changed, a clear reference back to the algorithms in the earlier design section must be given.

Please do not include any package-generated code. This was especially relevant for those candidates who had used the data source configuration wizard of VB.Net or the Express editions of the language to fill a dataset with data. This is effectively package-generated code to establish database connectivity and does **not** confirm complexity of the solution.

User Manual

The assessment of the User Manual is linked to the level of Complexity of the problem and its associated solution. This is the first year that a User Manual for the whole system, not just part of it, was required as discussed in detail during the 2010 COMP4 Teachers' Standardising Meetings. Some user manuals still failed to guide a user through all sections of the system.

A detailed table of contents for this section should be included. Not all candidates provided appropriate installation instructions for the different types of user/administrator of their system.

Nevertheless, most candidates incorporated screen shots with appropriate explanations, but perhaps should have considered the needs of their users in more depth. Some candidates had provided their user manual for the user to assess during acceptance testing. This was again good practice.

Error messages linked to screen shots, trouble shooting and recovery procedures generally needed to be more extensive. Sometimes, there was also very little data in the system when either testing it or writing the user manual.

Appraisal

Assuming that candidates had produced a detailed list of SMART objectives in agreement with their end user in the Analysis section, there was easy scope for comparison of outcomes versus objectives.

Also assuming that there really was a genuine end user, detailed feedback could easily be provided. Local assessors should clearly authenticate this feedback especially where it is obvious that the candidate has typed it out. Some of the feedback presented by candidates was unconvincing to say the least. The best feedback evidence appeared on headed notepaper, was signed and dated by the identifiable end user and had criticisms as well as praise for what had been achieved by the candidate.

Even if genuine feedback had been presented, many candidates still failed to analyse it and then use it as the basis for possible future developments/improvements. Many possible extensions (if indeed present) appeared to be entirely candidate driven.

Quality of Written Communication

This was usually quite good, but a detailed Table of Contents was not always present at the front of the report. Most centres accurately assessed this criterion, with the general view being that acceptable use of English, a well-structured report divided into the sections detailed in the specification and the appropriate use of word processing facilities could score 3 marks.

It is still disappointing that many candidates failed to use a word processor appropriately; many project reports were missing such basics as headers, footers and word processor-generated page numbers and a table of contents. Pagination was hand-written in some cases. It is not really appropriate to award full marks for this section when candidates put lots of important components of the documentation in appendices, especially if these are not accurately referenced in the sections where they should be situated. This was exacerbated if there was no overall Table of Contents at the front of the project or a fully completed Project Log Sheet. Minor spelling and grammatical mistakes that did not detract from the meaning were not penalised this year. A number of our candidates were ESOL (English for Speakers of Other Languages) candidates who often produced work of a high or very high technical standard.

Mark ranges and award of grades

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