COMPUTING

Paper 5216/01

Paper 1

General comments

The paper was of a comparable standard to recent sessions. The opinions of the marking team were that the candidates were, perhaps, slightly better prepared than they have been in the past. It is impossible to know why this should be, if indeed it is true, but there is the indication that Centres are using past papers sensibly in order to highlight particular areas of the syllabus rather than using the scatter gun approach. In other words, candidates are learning to interpret questions and, in their own minds, be able to decide which area of the syllabus the question has been set on and from there decide how to answer it. Too often, in the past, candidates have had to analyse the meaning of a question from the language in the question. If they can learn to recognise the part of the syllabus it is from then the answers fall into place. It is essential to the performance of candidates that Centres try to include this understanding of the syllabus, alongside the content, so that candidates see the syllabus as a homogenous whole rather than unconnected bullet points.

On the subject of bullet points, I have often said in these reports that the use of bullet points in responses is to be applauded, however, a few (thankfully very few) candidates have taken this as a challenge to make their answers as short as possible. One candidate answered the entire paper in bullet points, each of which contained a maximum of two words. Apart from the obvious problem of not being able to get points across properly, it becomes almost impossible to read it.

In similar vein, some Centres are being so successful with the use of past paper questions to illustrate and give practice in individual bullet points that the candidates are evidently taking their revision too seriously. We are beginning to see answers to some of the questions in previous years given as answers to similar questions from this year. This happens particularly when a candidate uses a different context in the second half of the paper.

Once again there were no time problems with the paper and I am delighted to say that there were no reports of any cultural issues which may have upset candidates arising from the questions. The extension of the paper, which had caused a degree of worry about what it meant for the paper content does not seem to have been an issue, for which I am most grateful.

Comments on specific questions

Question 1

(a) Well answered. The only problems were candidates who still didn't realise that for two differences you could not just give the two sides of one difference, and for the many Centres that are still teaching that the BIOS is stored in ROM. By definition, the BIOS is user definable, and consequently cannot be stored in a type of memory which does not allow any changes to be made. For the record it is usually stored in CMOS (RAM).

(b)(i) After many years we have relented and we have not penalised candidates who have used proprietary brand names. We would have liked to maintain the discipline of software 'types' but decided it was unfair on some candidates. Having said that, many candidates penalise themselves because if we are not familiar with the software that they name they are obviously not given the mark. It should be noted that many popular software packages are known by different names in different parts of the world, so the safest policy is still to stay with the 'no proprietary brand names' rule.

(ii) Too many candidates did not see the distinction between the CDRW and the DVD drives. Most candidates found great difficulty in coming up with sensible reasons for their choices.

Question 2

These two were very much Centre based as far as candidate success was concerned, particularly the form based interface which was often confused with a menu based interface.

Question 3

(a) Perhaps not the easiest question on the paper, that came later, but close to it. Most candidates gratefully accepted 6 marks here. Though there were a number of (Centre based) candidates who, rather bizarrely, decided to write down a solution to a problem and then try to think up a problem to go with it. Surely this sort of approach (though not penalising candidates as far as marks are concerned) can be nothing but confusing to a 16 year old. If there is one piece of advice that I think all of you who I have had the pleasure of meeting during various inset sessions around the world will agree with is that my main piece of advice is 'Keep things simple'.

(b) Well answered, particularly those who did not attempt to write an essay, but split up the salient points of their answers into bullets.

Question 4

(a) The only problem here was when a candidate did not give two differences but only one, although by saying the two sides of a difference they thought they had said two things.

(b) The only problem here was that many candidates said that a bus network was cheaper than any other sort of network. This is not necessarily true and they did not gain full credit for such a response. I have little doubt that the majority knew the correct answer.

(c) The definition is well known but the application to the problem of bit rate was very poorly answered. This idea of putting a part of a protocol in context and then asking for an explanation may well have confused some, more because of the difficult in phrasing the response than actually being unaware of the answer.

Question 5

(a) Candidates either knew about modularising or they didn't. Those who did scored most of the available marks.

(b) A difficult question, intended to be so. Many candidates described the ancestral filing system by simply giving the standard diagram. This did not answer the question. The emphasis is on the modularisation of the problem solution and the way those modules should fit together rather than a solution to the problem itself. The best solution is to draw a form of JSD and that is what appears in the mark scheme, however that was certainly not expected from the candidates and the marks were awarded for the logic behind the responses rather than for the style of answer.

This is the question that the extra time has been given for on the length of the paper. The answer is only worth 5 marks but my estimation is that for a candidate to make a good attempt at this question will involve them needing an amount of thinking time and probably time to do a draft version with various changes that have been felt to be necessary and then to produce a final version. It is disappointing to report that candidates treated this question in the same way as all the others, get the answer down on paper as quickly as possible. Candidates should practice this type of question because there will be one on the paper at each session (Not a JSD, but one that will require an application of logic. Could be an algorithm for instance) My advice is to teach them to have as good a go at it as they can and then go back to it at the end and go through the logic of what they have produced and change anything necessary.

Question 6

Standard bookwork, with most candidates scoring well.

Question 7

Earlier in the paper it was said that the easiest marks came later, here they are. Though, most candidates managed to say that a telephone number would be stored as an integer. Many countries around the world include leading zeros in numbers, or # keys or other symbols. Some have telephone numbers that do look like integers, so we fall back on the definition of an integer data type containing data with which it is possible to carry out meaningful arithmetic. The file sizing is being done well now and the acceptable field sizes were set to be very generous deliberately as these were meant to be available to every candidate. The archiving and back up routines which were intended to be rather more difficult and in the past would have caused problems, are now well understood and explained.

Question 8

(a) Far too general. Most candidates explained how verification and validation were carried out, that does not answer the question.

(b) Well answered.

Question 9

Part **(a)** has been asked before and was well answered here, partly because of the other question, but also because it is within the experience of the candidates. Candidates are not used to reading user guides though and the project guide that they write themselves does not come until next year, consequently the second part of the question which should have been an easy two marks proved to be anything but for most candidates.

Question 10

Very poorly answered. The distinction between the two areas of management information was ignored and most marks that were awarded were given for very general answers. Centres are advised to consult the published mark scheme for all the expected responses, but particularly for this question.

COMPUTING

Paper 5217/02

Paper 2

The new format for the Practical Programming Project (Paper 2) was examined for the first time this examination session. The programming project enabled candidates to show how well they could write programming code and its accompanying documentation. The weighting of the project is slightly lower than that of Paper 2 of the previous syllabus and it was hoped that it would not take up as much of the candidates' time as the tasks did.

It was really good to see the work produced from many Centres. They looked as though the candidates had learnt a great deal, and used their abilities to produce some excellent projects. Many of these were about 30 pages, with some very good programming projects being even smaller.

Since writing computer programmes is a major distinctive feature of the subject, it is hoped that Centres and candidates found the project to be worthwhile as it appears from what Moderators have seen.

Some Centres clearly had problems. At some, candidates wrote books, much of the content being repetitive and unnecessary. At other Centres, candidates produced system problems and appeared to be working to Paper 4 requirements. The major weakness, both in the development of the project and the marking, was the lack of code documentation, particularly annotation.

When considering the type of project and what needs to be taught towards it, the main point of reference is the 'Guidance on Marking the Practical Programming Project' on pages 22-24 of the 9691 syllabus. Centres are urged to ensure that they are fully conversant with the requirements of the programming project before deciding on possible projects and advising candidates.

The introduction should simply set the context for the work. A choice of topic should include a possibility of defining and manipulating records in some way. Simple numeric problems and games are usually not really suitable.

There are many design methods that can be used. From a perfect design, a third person should be able to follow the design and write the appropriate code to fulfil that design. The design then has to be turned into code –in any computer language.

Problems arose with the use of Visual Basic as an add-on to spreadsheet or database solutions. The marks available to candidates are for code that they have written themselves, so that the use of design wizards and autocode means that those sections of code are not available for marking. The candidate has not written those sections of code themselves.

The program code needs careful documenting. Code needs maintaining, updating and changing. This could be years later and by a completely different person. That person needs to be able to follow the code easily and relate it back to the design. There are several different ways towards this, which are all mentioned in the guidance. The main one though, is by annotation of the code. This is not just by a comment after the header of a procedure, but by a comment after most lines of the code.

Among this year's projects were several with excellent written code, but it was nearly impossible to know what it did. The code needs to be as transparent as possible. This was the main area of marking problems. With each of the eight specified programming skills there are two marks, one is for the use of that skill, and the other is for the annotation attached to that skill. Many of the Program Development marks depend on annotation as well.

The marking rules for Testing are fully laid out in the guidance. Where there was fewer than eight test runs it is impossible to gain more than four marks.

The reasons for the information required under Implementation are also quite specific.
The Moderators look forward to seeing more excellent projects and hope the above points are helpful.

COMPUTING

Paper 5218/03

Paper 3

General comments

The paper worked largely as expected, discriminating between candidates while at the same time allowing the more able candidates to show their knowledge and understanding as far as computer theory is concerned.

This was not an easy paper although there were some easier questions to ensure that the question paper was accessible to all candidates. Some of the responses were of the very highest standard and candidates who attained the highest grades are to be congratulated. However, congratulations should also go to all candidates who managed to achieve their potential and only the candidates concerned and their teachers can know who they are.

While there were a large number of very satisfying scripts to mark, there were an equivalent number of scripts submitted by candidates who were not prepared for the examination and for whom the time spent in the examination room must have been difficult.

There was no evidence of candidates getting into time trouble or that questions might include an element of cultural difference that could disadvantage some of the candidates. Over the last few sessions we have tried to tighten up on this and I am pleased to say that we seem to have reached a stage where the questions are considered to be fair to the candidates. There will always be occasions where a candidate is unable to answer a question, but if that is simply because they do not know the answer that is part of the examining process, the important thing is that cultural or language differences have not put them at a disadvantage.

Comments on specific question

Question 1

(a) The majority scored full marks here.

(b) Most had an idea about the differences but got tied up in the idea of the compiler translating all the code 'in one go'.

(c) Many candidates gave full answers about the stages of compilation and at what stage errors would be detected and reports given but that did not answer the question. There were relatively few who could satisfactorily say 'how' errors are recognised.

Question 2

(a) This was a well answered section of the paper, many candidates obviously taking their cue from the project that they have only just completed.

(b) While appreciating the need for the maintenance of a system, few candidates understood the different versions of maintenance and were reduced to answers of the form 'perfective maintenance makes the system perfect'. There were fewer still who were able to apply the different types to the situation of the supermarket system.

Question 3

This was an attempt to get away from the paper 1 type of question which asks for 3 reasons why simulation is sometimes used. In this question the candidates had to think of three aspects of a situation for themselves and then say why they would be simulated. Surprisingly most candidates were able to suggest three sensible aspects of the design but failed to give reasons why simulation would be needed.

Question 4

Again, a question that was in two parts. Candidates had to suggest four features of email and then state how each would be useful to the sales manager. There were some candidates who talked about cost and speed and efficiency, but thankfully very few. Most candidates were able to suggest features of email, but it tended to be only the better candidates who could explain their importance in this context.

Question 5

Most candidates were able to score full, or close to full, marks here. The difficulty is normally the stage between the MAR and MDR, though that is probably because of a difficulty with English rather than Computing.

Question 6

(a) It was rare to find a candidate who could demonstrate an understanding of the two programming paradigms. Some managed to score full marks, but the feeling was almost always that these candidates had learned definitions from a book rather than having an understanding of the concepts involved. Having said that, candidates do not have to use these languages beyond the rather simple questions set in some sessions and hence a deep understanding is an unreasonable demand.

(b) Most candidates were struggling here. Many thought that indexed addressing meant the use of hashing algorithms, It is difficult to keep these concepts separated in the stress of the exam room. The candidates who did understand these concepts sometimes fell down in the explanation. This is a question where a diagram is worth a thousand words and candidates who drew the two simple little diagrams scored the full marks for the explanation and invariably gained the mark for the reason as well. Teachers are encouraged to suggest the use of diagrams for many of the concepts in the syllabus. Candidates find them easier to understand in the main and they also overcome many of the problems with trying to explain a complex process in a second language.

Question 7

Some good answers here. All candidates were able to pick up some marks, though the explanation of the need for maintaining privacy was often phrased in generic terms rather than in the context of the health Centre.

Question 8

(a) Well understood by the majority of candidates, though many did use queue or stack as the example structure (for either dynamic or static) not understanding that stacks and queues are logical concepts and can be stored in arrays or linked lists.

(b) The required responses for this, and all other questions on the paper, are available in the published mark scheme. The answer in (ii) did not need to include a recognition that the search could end early in order to achieve full marks, but I do not recall seeing a single response that included this part of the algorithm, which was disappointing.

Question 9

(a) (b) This question was not answered very well. The intention was to make the marks more accessible for candidates by dividing the question into the five parts, however, the result seems to have been to make it more daunting than the standard questions about memory management. A common cause of misunderstanding is the difference between paging and segmentation. The simplest way to think about the two concepts is that paging is a physical concept so programs and data have to be squeezed into the fixed pages no matter how inconvenient, whereas segmentation is a logical way of dividing things up and consequently can adapt to the different sizes of appropriate divisions of the software and data although it makes the memory more difficult because there is no standard size. Candidates are still having problems with spooling. The concept which seems to cause the trouble is the idea of the spool queue containing references to the positions of the jobs (held elsewhere) rather than containing the jobs themselves.

Question 10

A final question which worked very well as a discriminator. All candidates were able to score some marks but only the best understood the ideas of standardisation of software and hardware and the reasons why they are needed.

PROGRAMMING PROJECT ADVANCED

Paper 5219/01

Paper 1

General comments

This report provides general feedback on the overall quality of project work for the Diploma in Computing candidates. In addition, all centres receive specific feedback from their moderator in the form of a short report that is returned after moderation. This reporting provides an ongoing dialogue with centres giving valuable pointers to the perceived strengths and weaknesses of the projects moderated.

Centres are again reminded that the programming project must involve the use of an object-oriented programming language and may also involve the choosing and installing of hardware. Centres are also reminded that candidates need to identify opportunities to develop and deploy a limited set (5–6) of library elements in their solution. Also the project work is designed to test the understanding of the systems life cycle, these requirements are clearly set out on pages 25 to 31 of the syllabus. The guidance on marking projects on pages 32 to 40 can also act as a useful checklist setting out the expected contents of each section.

The selection of an appropriate problem by the candidate is extremely important, as the analysis, design and implementation of a computerised system should always involve consultation with a user, ideally a 'third party' user throughout the development of the system.

Project Reports and Presentation

The presentation of most of the reports was to a very high standard, with reports word-processed and properly bound. However, the use of proof reading and a spell checker is to be recommended. Candidates are also reminded that the submission of magnetic or optical media is not required and the moderators do not consider it.

It is recommended that the structure of the report follows that of the mark scheme, this gives a clear outline as to contents for the candidates to consider and also aids the assessment by teachers and moderation of the work.

The use and development of library elements, set out in the separate sections required in the report, is essential to the object-oriented approach required for this component. Unfortunately, this session only one centre had ensured that their candidates had made good use of library elements and followed this approach.

Candidates can use library elements in different ways they can make use of pre-prepared libraries e.g. a library of date functions, they can identify new functions that the wish to use and either customise an existing library by adding new functions to it or set up a separate library of functions that is required for this particular system.

Project assessment and marking

Most assessment by centres was too generous, particularly where there was no evidence of user involvement and no use library elements were evident in the candidate's report.

Centres should use the mark scheme set out in the syllabus and include a detailed breakdown of the marks awarded section by section together with a commentary as to why marks fit the criteria set out in the syllabus. This greatly aids the moderation of the projects allowing moderators to identify why marks have been awarded.

The requirements are clearly set out on pages 42 to 51 of the syllabus in 'The Guidance on Marking the Computing Project' section. Also these requirements can also act as a useful checklist, for both teachers and candidates, setting out the expected contents of each section.

Centres are also reminded that candidates should use this guidance for the expected contents of their reports rather than some of the popular 'A' level textbooks available for project work, which do not cover the full requirements of the CIE syllabus.

Comments on Individual Sections

The comments set out below identify areas where candidates' work is to be praised or areas of concern and are not a guide to the required contents of each section.

(a) Definition, Investigation and Analysis

(i) Definition- nature of the problem

Most candidates described the organisation and some described the methods used but only the better candidates identified the origins and form of the data. Centres are reminded that a detailed description of the organisation covering many pages is not required here just a short paragraph covering the appropriate areas.

(ii) Investigation and Analysis

Candidates should clearly document user involvement and agreed outcomes. Better candidates clearly showed evidence of observation, interviews and investigation of documents currently in use. A detailed requirements specification based on the results of the candidate's investigation should be produced.

Also alternative approaches need to be discussed in depth and applied to the candidate's proposed system in order to obtain high marks.

(b) Design of the Library Elements

This section was not present in the majority of reports. It should include the following elements:

(i) Nature of the solution - A clear set of objectives with a detailed and complete design specification, which is logically correct. There are also detailed written descriptions of all processes/sections and a clear, complete definition of any data structures. The specification is sufficient for someone to pick up and develop appropriate library elements. The library elements have been designed to be reusable and easily configured.

(ii) Intended benefits of the library elements have been identified and explained.

(iii) Limits of the scope of the library elements.

(c) Software Development, Testing and Implementation of the Library Elements

This section was not present in the majority of reports.

(i) Development and Testing of the library elements - the examiner must be left in no doubt the library elements actually work in the target environment. Candidates should provide program listings in the form of printouts. Data structures should be illustrated as part of the listings where appropriate, detailing their purpose. There should be a full set of printouts showing input and output as well as data structures. All hardcopy should be fully annotated and cross-referenced. A full test plan, with evidence of each test run should be present in the report, together with the expected output for each library element. The test plan should cover as many different paths through the system as is feasible, including valid, invalid and extreme cases.

(ii) Appropriateness of structure and exploitation of available facilities used in the production of the library elements - some discussion of the suitability of methods used for the particular system should be included. Some recognition and discussion of the problems encountered and actions taken when appropriate should also be included. A log of such problems should be kept.

(d) Documentation of the Library Elements

This section was not present in the majority of reports. As many programmers work as part of a programming team, the documentation for the library elements is intended to allow the candidate to demonstrate their ability to work as a part of such a team.

(i) Technical - Much of the documentation will have been produced as a by-product of design and development work and also as part of writing up the report to date. However, a technical guide is a standalone document produced to facilitate easy maintenance and upgrade of a system. The contents of the guide should, where relevant, include the following: record, file and data structures used; database modelling and organisation including relationships, screens, reports and menus; data dictionary; data flow (or navigation paths); annotated program listings; detailed flowcharts; details of the algorithms and formulae used. Candidates should include a guide to the interface to the library routines - parameters, public and private data structures, formats etc. All parts of the guide should be fully annotated since this is important for subsequent development of the system. The specifications of the hardware and software on which the system can be implemented should be included.

(ii) Clear guidance, as friendly as possible, should be given to allow the incorporation of the library elements in other solutions. Details of the public interface should be provided for each of the library elements. Some mention here of the relationship between the elements and the data they deal with may be relevant. The user guide should be well presented with an index and, where necessary, a glossary of the terms used.

(e) Design of the main solution

(i) Nature of the solution

Centres are again reminded that the requirements specification set out in the analysis needs to be discussed with the user leading to a set of achievable, measurable objectives that have agreed with the user. These objectives will then form the basis for the project evaluation. Candidates often clearly set out proposed data structures and designs for input screens but then forgot to provide a detailed description of the processes to be implemented and designs of the required outputs.

(ii) Intended benefits

Candidates need to clearly identify the merits of the intended system.

(iii) Limits of the scope of solution

Candidates need to discuss the limitations of the intended system and estimate the size of the files required.

(f) Software Development, Testing and Implementation of the Main Solution

(i) Development and Testing

Evidence of testing needs to be supported by a well designed test plan that includes the identification of appropriate test data, including valid, invalid and extreme cases, and expected results.

(ii) Implementation

Few candidates included an implementation plan. This should contain details of user testing, user training and system changeover that have been discussed and agreed with the user. These details need to be clearly related to the candidate's own project not discussed in general terms.

Evidence of user testing is essential if high marks are to be awarded for this section. Better candidates included photographs of the user testing the new system, printouts of the testing together with signed comments from the user and/or a letter from the user commenting on the tests and their results.

(iii) Appropriateness of structure and exploitation of available facilities

Candidates should discuss the suitability of both hardware and software at this stage. Few candidates kept a log of any problems encountered together with details of how these problems were overcome. Any system developer encounters problems; these problems need to be noted together with the corrective action taken.

(g) Documentation of the Main Solution

(i) Technical

Very few candidates produced a stand-alone technical guide including the following: record, file and data structures used; database modelling and organisation including relationships, screens, reports and menus; data dictionary; data flow (or navigation paths); annotated program listings; detailed flowcharts; details of the algorithms and formulae used. Candidates need to annotate all parts of this guide since this is important for subsequent development of the system. The specifications of the hardware and software on which the system can be implemented should also have been included.

(ii) User

For full marks the candidate needs to include an index and a glossary, the guide needs to be complete including details of backup routines and common errors. Also good on-screen help should exist where this is a sensible option.

(h) Evaluation

This section is very poorly completed by many candidates, with many trying to attempt an evaluation without evidence provided from their end users. End user involvement is clearly required in (i) and (ii) of this section. There are detailed guidelines, for this and all sections, clearly set out in the guidance for marking projects section of the syllabus.

(i) Discussion of the degree of success in meeting the original objectives

Very few candidates considered each objective in turn and indicated how the project met the objective or explained why the objective was not met. Even fewer candidates included use of user defined, typical test data as part of this discussion.

(ii) Evaluate the users' response to the system

Many candidates did not provide clearly recorded evidence from their end user, but this is essential. Candidates need to obtain the users' response to how the system developed meets the agreed specification and evaluate this response as to the satisfaction with the system developed.

(iii) Desirable extensions

Many candidates identified limitations and possible extensions but sometimes forgot to identify the good and bad points of the final system.