COMPUTING

Paper 9691/11
Written Paper

Key message

As this subject moves to a syllabus where more understanding and application of the syllabus topics rather than just simply learning definitions is required, candidates will need to change how they approach computing topics. This is a challenge which many Centres and candidates, of course, will relish.

General comments

It was evident that a significant number of candidates are learning certain topics "parrot fashion". Questions which required an application of knowledge were overall less well answered.

Candidates and Centres are reminded that written papers are now scanned in and marked on computer screens by Examiners. Consequently, if a candidate writes the answer to a question on an additional page they must indicate VERY CLEARLY to the Examiner where their revised answer is to be found. If answers are "scrubbed out", the new answers must be very clear so that Examiners can easily read the text and award candidates the appropriate mark.

Comments on specific questions

Question 1

(a) This was generally answered reasonably well. Several candidates gave examples here which did not really describe the *purpose of storage devices* and consequently no credit could be given for these examples.

(b) Many candidates correctly chose floppy disk and memory sticks (flash drives) as examples of suitable storage devices.

(c) In part **(i)**, candidates usually answered "one user at a time" and in part **(ii)** "use a word processor, e.g. at the same time as surfing the net." Very few other points were raised.

Question 2

(a) This was not very well answered at all with many candidates giving very general answers. The only real points mentioned were: "consideration of hardware and software" and "objectives were agreed". Most answers covered very general points such as analyst and client discussions, the problem with the existing system or the analyst watching the process.

(b) Again this was not well answered. Many candidates just spent the whole answer discussing changeover methods (direct, parallel, phased and pilot). Very few mentioned installation of hardware and software, conversion/transfer of files or the need to retrain staff.

Question 3

Many candidates used the published mark scheme from a similar type of question in June 2011 and tried to apply it (unsuccessfully!) here. This question referred to a car navigation system (sat nav) and needed reference to, for example:

- touch screens (for menu options)
- speech output (safety feature)
- need for as little information as possible (small screen)
- clear colours to show routes

Question 4

This question referred to setting up and using an *expert system.* There were too many "throw away answers" such as "create the data base", "create a rules base", "use a search engine" or "make an interface". At this level, an understanding of how *expert systems* work and the rôles of the knowledge base, inference engine, rules base and explanation system is expected.

Question 5

(a) In part **(i)** many candidates made reference to data type (used in the array) or the array as a kind of list. Part **(ii)** could only be reasonably answered using some form of code, such as:

```
INPUT item
FOR i = 0 TO x
        IF array [i] = item THEN "found"
NEXT i
REPORT "not found"
```

If coding was not offered, then the candidate would need to cover the following points: correct looping structure, a way to identify if item is found, produce an error message if item not found after all checks and some conditional statement to ensure all array items checked.

(b) Candidates made reference to a FIFO structure or to use of pointers. More detail (such as 2 pointers used, check if queue is full or empty, for example) is needed here.
Surprisingly, very few candidates used diagrams to help in their descriptions. Questions of this type are usually much easier to answer if a diagram (or diagrams) is included to embellish the description.

Question 6

(i) This question was, in general, well answered with sensors, keypad and remote control device being the most common input devices. Weaker candidates mentioned a thermometer rather than temperature sensor.

(ii) This part of the question gave few problems. However, some candidates said heaters and fans were a type of output device. The most common correct answers were LCD screens to show the current/set temperature or beepers/loudspeakers to indicate an error or that the changes to the settings were accepted.

Question 7

(a) Part **(i)** was well answered with many candidates understanding the features of a form-based interface.

In part **(ii)** many candidates made reference to faster data access. The better candidates also referred to the need for fast access to enable rapid responses to customer queries.

(b) Most candidates understood the need for back-up and how the procedure is carried out. There were some unrealistic frequencies of back-up offered, such as every hour or every 3 months, for example.

Question 8

(a) This part of the question was answered well.

(b) This was not generally understood with many candidates just describing the need for a fast bit rate and what was meant by bit transfer rate. Very few linked this to on-demand video streaming or real time (true) video streaming and how the bit rate affected both of these systems.

(c) In part **(i)**, many candidates gave one example (e.g. parity type, serial or parallel data transmission) but very few mentioned agreement on character set or handshaking to ensure that both parties are ready to communicate.

In part **(ii)**, candidates need to understand that to allow different devices to communicate effectively, different layers deal with different parts of communication. If one layer is to be altered there is no need to alter other layers, only the links between the layers need alteration.

(d) This question caused few problems with the majority of candidates understanding what is meant by off-the-shelf software.

Question 9

(a) This was well answered by most candidates.

(b) This was well answered by most candidates.

COMPUTING

Paper 9691/12
Written Paper

Key message

As this subject moves to a syllabus where more understanding and application of the syllabus topics rather than just simply learning definitions is required, candidates will need to change how they approach computing topics. This is a challenge which many Centres and candidates, of course, will relish.

General comments

It was evident, however, that a significant number of candidates are learning certain topics "parrot fashion". Questions which required an application of knowledge were overall less well answered.

Candidates and Centres are reminded that written papers are now scanned in and marked on computer screens by Examiners. Consequently, if a candidate writes the answer to a question on an additional page they must indicate VERY CLEARLY to the Examiner where their revised answer is to be found. If answers are "scrubbed out", the new answers must be very clear so that Examiners can easily read the text and award candidates the appropriate mark.

Comments on specific questions

Question 1

(a) This was reasonably well answered with the majority of the candidates being able to adequately define input and output devices.

(b) Again reasonably well answered. Weaker candidates did not give the required description of the actual purpose of the device e.g. "bar code readers are used to input the bar code" as a description was inadequate. The answer needed to identify that the barcode reader inputs the id number which is stored in the barcode itself, for example.

(c) This question was not very well answered. A large number of candidates were unable to relate this to a time sharing system (i.e. reference to time slices, round robin system or use of priorities, for example). Many thought this was a simple network and described a typical network system.

Question 2

(a) This was not very well answered at all with many candidates simply discussing questionnaires, interviews (etc.). Many others just described the systems life cycle. Analysing the requirements of a system involves the collection and analysis of information about the problem, creation of diagrams (such as DFDs), consideration of different methods of solution, setting of objectives, etc.

(b) (i) This was basically well answered, but it was common to read: "the old system is completely replaced by the new system". This is true, eventually, of all implementation methods. There was a need to indicate the immediate changeover (e.g. old system stops being used one day and the new system is used the next day).

(ii) Phased and Pilot implementation methods were confused by many candidates. Several gained no marks for descriptions such as "system introduced department by department" which is essentially Pilot implementation. Phased implementation involves one part of the new system being introduced and other parts only when the first part is proved to be effective.

Question 3

Generally well answered with many candidates referring to bright colours (to gain young children's attention), use of animation and use of sound. Several mentioned the feature but did not describe why the feature was chosen.

Question 4

This question referred to setting up and using an *expert system.* There were too many "throw away answers" such as "create the data base", "create a rules base", "use a search engine" or "make an interface". At this level, an understanding of how *expert systems* work and the rôles of the knowledge base, inference engine, rules base and explanation system is expected.

Question 5

(a) (i) Many candidates mentioned the use of a common name, arrays being made up of rows and columns, for example. The better candidates mentioned that it can store a set of data items of the same type and that they would be stored in consecutive memory locations.

(ii) To answer this question successfully, candidates really needed to include some form of coding; for example:

```
FOR i = 0 TO length
        FOR j = 0 TO breadth
                array [i, j] = 0
        NEXT j
NEXT i
```

If coding was not supplied then the candidate needed a good description of how the array elements could all be initialised to zero.

(b) Candidates made reference to a LIFO structure the use of pointers. More detail (such as 2 pointers used, check if stack is full or empty, for example) is needed here.

Surprisingly, very few candidates used diagrams to help in their descriptions. Questions of this type are usually much easier to answer if a diagram (or diagrams) is included to embellish the description.

Question 6

Candidates need to understand that CDs are not magnetic media and DVDs/CDs are not solid state systems. Hard drives were also sometimes chosen as examples of solid state memories. There is probably some confusion here due to hard drives being replaced by solid state memories in many new laptop computers.

Question 7

(a) (i) This part was well answered with many candidates understanding the features of a form-based interface.

(ii) Some candidates did not realise that this referred to hashing algorithms and gave long descriptions of random access files with no mention of hashing algorithms.

(b) **M**ost candidates understood the need for archiving and the reasons why it is carried out.

Question 8

(a) There were no problems to report with this part of the question.

(b) There were some good answers from candidates who described how check sums were carried out and the reasons why they are used. Unfortunately, a surprising number described a parity check instead and discussed how the number of 1s were summed to find out if the byte was even or odd.

(c) There were no problems to report with this part of the question.

(d) Very general answers were given of how a stock control system would work. Many did not realise the role of barcodes or barcode readers and wrote about alarms, warning the manager of low stock and some even suggested that warnings would be given when stock levels reached zero. Candidates need to understand that stock entering or leaving the warehouse is identified using barcodes. These are used to find the item details on the system and update the stock amount. The stock level is compared against the **minimum** stock level and if less then a re-order flag is set. A regular check for re-order flags is done to create orders.

Question 9

(a) This was well answered by most candidates.

(b) The better candidates realised that the logic circuit was a *half adder.*

COMPUTING

Paper 9691/13
Written Paper

Key message

As this subject moves to a syllabus where more understanding and application of the syllabus topics rather than just simply learning definitions is required, candidates will need to change how they approach computing topics. This is a challenge which many Centres and candidates, of course, will relish.

General comments

It was evident that a significant number of candidates are learning certain topics "parrot fashion". Questions which required an application of knowledge were overall less well answered.

Candidates and Centres are reminded that written papers are now scanned in and marked on computer screens by Examiners. Consequently, if a candidate writes the answer to a question on an additional page they must indicate VERY CLEARLY to the Examiner where their revised answer is to be found. If answers are "scrubbed out", the new answers must be very clear so that Examiners can easily read the text and award candidates the appropriate mark.

Comments on specific questions

Question 1

(a) This was reasonably well answered with the majority of the candidates being able to adequately define input and output devices.

(b) Again reasonably well answered. Weaker candidates did not gain give the required description of the actual purpose of the device e.g. "monitors are used to show customers information" as a description was inadequate. The answer needed to identify that the monitor could show item prices or the total bill, for example.

(c) This question was reasonably well answered. The majority of candidates understood how batch processing worked.

Question 2

(a) This was not very well answered at all with many candidates simply discussing questionnaires, interviews (etc.). Many others just described the systems life cycle. Analysing the requirements of a system involves the collection and analysis of information about the problem, creation of diagrams (such as DFDs), consideration of different methods of solution, setting of objectives, etc.

(b) (i) This was basically well answered with most candidates understanding parallel implementation.

(ii) Phased and Pilot implementation methods were confused by many candidates. Several gained no marks for descriptions such as "system introduced part by part" which is essentially Phased implementation.

Question 3

The use of a user interface in an industrial application (power generating plant) caused problems for many candidates who found it difficult to transfer their knowledge of user interfaces to this type of scenario. The important aspects are: touch screens, use of colours to show OK/fault, alarms, constant safety checks (such as "are you sure you want to turn off that valve?") and so on

Question 4

This question referred to setting up and using an *expert system.* There were too many "throw away answers" such as "create the data base", "create a rules base", "use a search engine" or "make an interface". At this level, an understanding of how *expert systems* work and the rôles of the knowledge base, inference engine, rules base and explanation system is expected.

Question 5

(a) To answer this question successfully, candidates really needed to include some form of coding; for example:

```
FOR i = 0 TO end
        array [i] = 0
NEXT i
```

If only a description was given, the candidate needed to cover each point such as description of a correct loop, description of loop conditions and how each array element was set to zero.

(b) (i) Candidates made reference to a LIFO structure or to use of pointers. More detail (such as 2 pointers used, check if stack is full or empty, for example) is needed here.

Surprisingly, very few candidates used diagrams to help in their descriptions. Questions of this type are usually much easier to answer if a diagram (or diagrams) is included to embellish the description.

(ii) As in part **(a)**, this question is best answered using coding; for example:

```
IF pointer + 1 > size of array
        THEN report stack is full
                ELSE continue with the operation
```

If only a description was given, the candidate needed to include a description of any conditional statements needed, how to carry out check to see if pointer is larger than stack size and some indication that the stack is full or carry on to add another value to the stack.

Question 6

This question caused very few problems to the majority of candidates. However, a surprising number of solid state and magnetic media seemed to appear here. Candidates need to understand that a CD-ROM is not suitable as a way of collecting data to be used on their home computer.

Question 7

(a) (i) This part was well answered with many candidates understanding the features of a menu-based interface.

(ii) This part of the question was not very well answered with very few understanding why sequential access was used.

(b) This question also did not cause any real problems with most candidates understanding the need for back-up and the reasons why it is carried out. Although there were some "unusual" back-up frequencies given such as every 2 hours or every 3 months.

Question 8

(a) There were no problems to report with this part of the question.

(b) The concept of parity checks was well known by the majority of the candidates. Although how parity checks could be used to detect errors was less well understood. Very few mentioned how the parity check was used at the receiving end and even fewer mentioned parity block checking.

(c) (i) This part caused no real problems.

(ii) Only the better candidates knew why a range of values was used to control the thickness of the sheets. Weaker candidates struggled to understand the concept of the measurement being analogue and therefore difficult to be totally correct. They also found it difficult to understand that the computer would have to make constant changes if a fixed value (e.g. 5.0 mm) was used.

(iii) Likewise, few candidates understood the role of sensors, actuators and computers when trying to control the process through feedback (real time systems).

Question 9

(a) This was well answered by most candidates.

(b) This was well answered by most candidates.

COMPUTING

Paper 9691/21
Written Paper

Key message

Candidates need practical programming experience to do well in this component.

General Comments

This was the first time that this paper had been sat in the November examination series.

Many candidates produced very good answers, particularly for the questions that involved writing programming code. This was in contrast to the candidates who could not answer the coding questions, and who seemed unsure what language they might be using. This appears strange since the paper is testing candidates' knowledge of programming and it is assumed that they will have written, debugged and tested meaningful programs.

The candidates who performed best in this examination were those who had written programming code and used it to solve a problem. They were the ones who had detailed knowledge of a language and knowledge of the processes involved in solving a problem.

Comments on specific questions

Question 1

(a) Nearly all candidates could follow the flowchart correctly.

(b) Most candidates got some of this question correct. The main weaknesses were in the input and output statements where some candidates used instructions that are not in the language specified. This indicates that those candidates had probably not used the correct instructions.

(c) A lot of candidates wrote that they had used iteration when they had not.

(d) Most candidates had a vague idea of what iteration is, but few could be specific.

(e) All candidates had a good try at this question. The main weakness in the answers was that their counter variable was not initialised. This meant that it was not possible to tell if the ending condition was correct.

(f) One of the best answered questions.

(g) This was well answered. Candidates were quite able to find and correct the error.

(h) Most candidates were able to get some of this coding correct. Usually, though, variables were incorrectly initialised and few could write the WHILE loop without errors.

Question 2

(a) A lot of candidates wrote general responses to this question. It needed to be related to the scenario.

(b) and (c) Both questions were well answered.

(d) Many candidates knew about local variables, but were not able to describe the scope of a local variable.

(e) The response to **(i)** by many candidates was correct, but they were usually unable to give any rules in their chosen language.

(f) Generally well answered.

(g) Most candidates answered the first part correctly. The second part was fairly well answered. It was difficult to see why many candidates gave invalid data when the question excludes that. Also how candidates thought that 2 items of invalid data could be tested together.

(h) A lot of candidates did not deal with each condition separately, which seems to indicate that they had not used a compound condition in a program**.**

Question 3

(a) A lot of candidates struggled with the trace table. Few could pick out the individual letters indicated.

(b) Only a small number realised that the string was being reversed.

(c) To answer this well the candidates had to realise that surprise was the name of the function, so assigning a value to it had a particular effect.

(d) This was poorly answered, given that procedures are a main building block of procedural languages.

(e) (i) Generally good responses.

(ii) Those candidates who answered this part only wrote in English what the code does. Annotating code needs explanation of the **effect** of the code.

(f) Good responses from those who knew that characters are compared in turn from the left hand side and the first higher code value determines the largest word. If two words are the same when one ends then the other is the larger alphabetically.

COMPUTING

Paper 9691/22
Written Paper

Key message

Candidates need practical programming experience to do well in this component.

General comments

This was the first time that that this paper had been examined in the November examination series.

There was a full range of marks, with many very good candidates. The impression that came from their papers was that they had done a considerable amount of programming, had applied their knowledge to a problem, and had been through all the associated tasks such as debugging and testing their programs.

Some candidates gave correct solutions to the questions involving segments of code, but many gave answers that looked as though they knew of the programming structure, but had never used it in the harsh light of a program that they want to work.

As can be seen from the detailed comments, it is difficult for a candidate to do well on this paper if they have not programmed to solve a problem. They need to see the whole process through to know the techniques and coding detail to the level expected.

Comments on specific questions

Question 1

(a) All candidates gave a response to this question. Many were very general and bore no relation to the scenario it was set in.

(b) and **(c)** Well answered by nearly all candidates.

(d) Whilst a good proportion of candidates knew how to answer this question, a large number appeared to know nothing about local variables and their scope.

(e) (i) Many candidates were able to answer this question sensibly.

(ii) Even more were able to give an example from a high-level language. It was good to see that algorithm and pseudocode were absent as high-level languages this time, but machine code was wrongly quoted.

(iii) Few candidates could give three rules. Most candidates gave one correct response, that of not starting with a number.

(f) The numerical parts were well answered; but few could state that the testing strategy to find such errors is black box testing.

(g) (i) Most candidates could get at least one of valid/normal and boundary/extreme data.

(ii) The Examiners were happy to accept 12 and 24 as either normal data or boundary data. Many candidates realised that with only three possible values they all had to be tested. It was difficult to see why some candidates gave invalid data when that was specifically excluded in the question, nor how they hoped to be able to test two pieces if invalid data at the same time.

(h) A lot of candidates did not deal with each condition separately and then link it with a Boolean operator, which seems to indicate that they had not used a compound condition in a program**.**

Question 2

(a) Most candidates followed the flowchart correctly.

(b) The responses to this question were disappointing. The flowchart had a CASE statement in it, but many candidates chose to use IF statements. Those who did use a CASE or SELECT statement rarely knew the syntax of their chosen language.

(c) About a half of the responses said that iteration had been used when it had not.

(d) All candidates had heard of iteration, but few could explain what the term means.

(e) Most candidates could add some correct boxes to the flowchart, and usually at a correct position. The box that was often not present was the initialisation box. This meant that it was difficult to know if the condition in the decision box was correct.

(f) Generally well answered.

(g) One of the best answered questions on the paper. Candidates were able to correct code well.

(h) This question received very poor responses. A FOR loop is a basic building block. Nevertheless candidates used a FOR with a WHILE on the same line. They incremented the loop counter inside the loop, and often used an identifier for the loop counter, but a different identifier in the array suffix. The responses suggested that only the best candidates had experience of using a FOR loop controlling an array in a meaningful piece of programming.

Question 3

(a) the trace table was not completed well by most candidates. Picking out the letters seemed too much for many candidates.

(b) Those candidates who followed the trace correctly often realised what the function was doing; i.e. coding each letter to the next letter.

(c) Only a few candidates answered in terms of the function name.

(d) Not many candidates could write about the features of a procedure, even though it is a main building block of procedural languages.

(e) (i) Usually well answered.

(ii) Candidates were asked to give a description of the 2 lines of code as if it were annotation. Many, however, just wrote the 2 lines in words when they should have been describing the action of the 2 lines so that someone else could understand what the effect of the lines of code was: for example: the first line evaluates the ASCII value of the current character and increments it by 1, i.e. gets the next letter alphabetically. The second line puts the new character in current letter position.

(f) Those who knew the answer to this question answered it well; i.e. it joins two strings together.

COMPUTING

Paper 9691/23
Written Paper

Key message

Candidates need practical programming experience to do well in this component.

General Remarks

This was the first time that this paper had been set for the November series of examinations. It was generally answered well. The best candidates were those who were able to write pieces of code accurately. They were also the candidates who knew most about the tasks surrounding writing a serious program, such as testing and debugging.

Candidates taking this paper generally answered the programming code questions well, and with this ability did well on the paper.

Comments on specific questions

Question 1

(a) There were many general responses, but to gain the mark the answer needed to be relevant to the question's scenario.

(b) (c) Well answered.

(d) Many candidates knew that the answer involved local variables. Not many were able to add to that, such as mentioning the scope of a local variable.

(e) (i) Most candidates taking this paper could answer this part.

(ii) Most candidates responded with at least one rule.
(iii) Well answered.

(f) (i) (ii) Both parts were well answered.

(iii) Very few candidates knew that the answer was Black Box.

(g) Both parts of this question were answered quite well. Only a few candidates gave the unwanted invalid values.

(h) (i) (ii) Many candidates did not keep each condition separate in either part and combine it with logical ORs.

(iii) Most candidates had the right ideas to draw a good screen.

Question 2

(a) All candidates could follow the flowchart correctly.

(b) This piece of code was mainly written well. The weakest part of the responses was using an input instruction that was not a part of the language that the rest of the code was written in.

(c) Generally well answered.

(d) Most candidates knew that iteration involved repeating something, with few candidates being sure just what.

(e) Most candidates answered part of this question. The main omission was in not initialising their counter variable. This meant that the Examiners could not determine whether the condition to end that sequence was correct or not.

(f) Most candidates were confident of their responses to this question.

(g) Correcting an error in the logic was the best answered question on the paper.

(h) The response needed appeared to be a straightforward block of code. Many candidates added an increment to their counter variable at the end of the loop, within the FOR loop. Others used a FOR WHILE construction. This suggest insufficient practice of basic programming constructs in a real programming language.

Question 3

(a) This trace table was answered well.

(b) Most candidates who followed the trace correctly realised what the function did, i.e. pick out the first word of a group of words.

(c) Surprise was the name of the function so the answer needed some reference to the use of the function name.

(d) As a function is a main building block of procedural languages it was surprising that so few candidates could write so little about it.

(e) (i) A comment or annotation within code would explain what an instruction does. Candidates usually wrote the line of language in English, rather than explaining it. For example: this line ends the REPEAT statement by finding an empty space indicating the end of the word.

(ii) Generally answered well.

(f) Very few candidates could answer this question. A good answer would have been: characters are compared in turn from the left hand side. The first higher code value determines the largest word. If two words are the same when one ends, the other is the larger alphabetically.

COMPUTING

Paper 9691/31
Written Paper

General

Performance across the various syllabus content was patchy with some areas well answered by the majority of candidates but other areas where there was no real understanding demonstrated. Surprisingly, content such as **Question 6** has been examined on many previous papers yet candidates' understanding was way off track and rarely scored more than 2 of the available 6 marks. Very few candidates had any idea about the drawing of a syntax diagram (**Question 10b**) or the use of the FAT table (**Question 1a**) by the operating system (this is specifically mentioned in the syllabus). Candidates need to be reminded that there is no question choice on this paper and so a failure to study certain syllabus content is a very risky strategy for success.

Candidates also need to appreciate that many of the computing topics they have met through their practical high-level language programming will be relevant for this paper (for example, **Question 8**).

Question 1

(a) Very poorly answered by the majority of candidates. The term 'cluster' was rarely seen and candidates were usually only able to state that the FAT was used to manage the files on the backing store. Some candidates confused the use of the FAT and the disk directory, wrongly suggesting that the FAT contained details about each file such as its file type, date last saved, etc.

(b) This was better answered but candidates rarely scored the full three marks.

Question 2

(a) Well answered. Candidates correctly stated this was effectively the stored program concept and gained further marks with an expansion about a single processor, shared memory and program instructions and data being indistinguishable.

(b) (i) Well answered with the role of the program counter well understood.

(ii) Candidates often only stated that the register stored the current instruction and did not then expand on this explaining how the instruction needed to be decoded into its op code and operand parts. If the operand is an address then this must be communicated to the MAR.

Question 3

(a) Generally well answered with candidates confident with a sign-and-magnitude representation.

(b) For many candidates this was straightforward and they were able to carry this understanding through in order to answer part **(c)**.

(c) Candidates who gave incorrect representations for part **(a)** and carried these through for use in part **(c)** were not penalised and credit was given where the candidates could correctly demonstrate binary addition. The instruction in the question stem *"You must show your working"* should have conveyed to the candidates they should show the carry bits used after each pair of digits and also indicate what was to be done with any final carry bit which resulted.

Again there was no penalty for the candidate who started with incorrect bit patterns carried over from part **(b)**. Most candidates realised that the sum to be performed was effectively (+93) + (-69).

Question 4

(a) (b) Well understood (probably due to the candidate's own experience of the use of an Intranet at their School/College). Points expected for **(b)** were sometimes given in part **(a)** and the marking was flexible in giving credit where possible. Stronger candidates gave answers which focused on the 'computing' issues, describing a network which used Internet protocols or content which consisted of web pages made available from a local web server viewed using web browser software.

Question 5

(a) Candidates generally scored well and were able to name an input device – typically a pressure or infra-red sensor and an output device. The most popular output device was a speaker for the signalling of some form of error, although the most obvious answer expected was an actuator to trigger the movement of the robotic arm. Candidates should note that 'sound' or a 'beep' is not an output device.

(b) This should have been a straightforward question but candidates often gave trivial or duplicate answers. Points which gained credit were: Can work 24/7, produces the product to a consistently high standard, robots do not require the ancillary facilities that human workers require, robots can work in conditions humans would deem hazardous. Points which fell short included; robots do not get bored, robots work faster than humans, robots are more efficient and many others.

(c) Very few candidates scored the full three marks. Answers expected were the writing of a new program, the amendment of the code for the existing program or the supply of new parameters to the existing software.

Question 6

For a topic which has been frequently examined, answers here were extremely poor. Whatever the candidates interpretation of how interrupts are managed there should have been at least four straightforward marks; the current process is suspended, the source of the interrupt is identified, the interrupt will have its own Interrupt Service Routine code (ISR) and at some stage the interrupted process will be continued. Candidates seem obsessed with the idea that interrupts have priorities and this is used to somehow queue up a number of interrupts. Most systems 'mask out' all interrupts of a lower priority once an interrupt is received and this concept does not appear to be understood by candidates. It was very rare to see scripts which scored more than 2 of the possible 6 marks.

Question 7

(a) Candidates were able to produce a description of a relational database but were less forthcoming with a description for a flat-file database. Expected was "A database consisting of only one file of records where all records have the same structure".

(b) There were many correct advantages stated by candidates. Most popular were that: data repetition/duplication will be minimised – data retrieval is done using 'point and click' queries. Only the better candidates appreciated that changes to the data structure of a table can be made without effecting the working of existing applications programs which access the table data (i.e. 'program-data independence').

(c) Well answered. Candidates appreciated the question was specifically about measures which could be implemented with the relational database/DBMS software; hence answers which described physical locks on the door, etc. were not relevant.

Question 8

(a) (i) (ii) For candidates with a minimum of practical programming experience that should have been two easy marks; however the keyword 'scope' was rarely used to illustrate their answer.

(iii) (iv) Less well understood; with the answers for **(iv)** rarely describing the address of a variable being passed through the procedure interface. Some candidates gained credit by stating that a value which is changed during the procedure code would result in the change being retained by the calling variable if passed by reference, but not so if passed by value.

(b) This was poorly understood. Candidates were often vague about what was actually being passed describing a 'value' or even the 'procedure itself'. This is a straightforward concept: the stack must store the values of any parameters which are passed and the return address. Rarely seen was an answer which described nested calls and how they would be managed.

Question 9

(a) Candidates generally scored well with a description of the tokenisation of the source code and the construction of the symbol table. Answers for part **(ii)** were less clear. What was expected was a simple statement that the construction of the program statements is compared to the formats which are permitted as described by the rules of the language.

(b) (i) This should have been an easy two marks for any candidate with practical experience of saving code to a module which is then made available to future programming projects.

(ii) Answers often followed from **(i)** describing the need to link existing library routines to the current program. The loader was well understood as the software which must manage memory addresses for the various code.

Question 10

(a) Most candidates scored well with a wordy explanation rather than a formal attempt at drawing a parse tree.

(b) Drawing a syntax diagram was alien to most candidates. Candidates need to be familiar with this. This particular diagram illustrated all the key points candidates are likely to meet; the need for a single entry and exit point, the importance of components in the correct order and how recursion is represented.

COMPUTING

Paper 9691/32
Written Paper

General

Performance across the various syllabus content was very varied. Particularly weak areas appeared to be the poor understanding of the special purpose registers used in the fetch-execute cycle and the role the data bus and address bus play in this (**Question 2**). Another general weakness was candidates' ability to decode the bit pattern for a floating point real number (**Question 3 b (iii)**).

Question 1

Most candidates were able to describe a queue data structure for the management of print jobs. It was intended that points in **(a)** be more accessible marks with the concepts that spooling came about due to the speed mismatch between the processor and input/output devices and that print jobs have to be temporarily stored before they are output.

Question 2

(a) (i) The question asked how the contents of the Memory Address Register change and candidates generally showed some understanding that its role is in storing the address of the next instruction to be fetched and that this memory address is copied to it from the Program Counter. There were few answers which went on to describe how the MDR is involved after the instruction in the Current Instruction Register has been decoded and the operand was an address which now needed to be accessed, hence involving the MDR.

(ii) Very poorly answered. Candidates had little idea how the Index Register is used in indexed addressing and candidates often confused the IR with the CIR. Some answers described its contents being incremented (which was not relevant to this question), or its contents being added to a base register value. Candidates do need to be exposed to some assembly language programming - probably using simulator software - to fully gain understanding and application of this syllabus content.

(b) Answers were often vague, typically describing no more than *"the address bus uses addresses"* and the *"data bus uses data values"*. Better communication was needed here with answers describing an address being 'carried' on the address bus gaining credit, whereas an answer describing values being 'stored' did not; similarly for the data bus description.

Question 3

(a) (i) Generally well answered.

(ii) Well answered by the stronger candidates. Candidates who showed their working often gained one or two of the available marks following (typically) an incorrect division by 16.

(b) (i) (ii) Candidates often gained the marks for **(i)** but then failed to score for **(ii)** suggesting that candidates are more confident in dealing with positive numbers.

(iii) Generally poorly answered. The number of scripts gaining the full four marks were very few. Candidates often dealt correctly with the exponent to gain one mark, but then could not compute the mantissa or shift the bits in the mantissa to produce a final answer.

Question 4

(a) Answers rarely progressed past a statement about the danger of hackers. Stronger answers which went on to suggest that the data could be used for purposes which were not in the patient's interest were rare.

(b) Measures which could be taken were well understood by candidates. However, the key word in the stem of the question was 'describe' and so candidates who produced only a list of five measures scored fewer marks.

Question 5

(a) The most popular answers which gained credit for an input device were a pressure sensor or an infra-red/light sensor. The explanation given was for the detection that the two parts were in place ready for assembly, or – for a pressure sensor – that the sensor could detect that the parts had been assembled to the correct torque. Answers for the output device were weak and surprisingly few candidates gave an actuator, using instead a loudspeaker to indicate some error had occurred. Candidates should note that 'sound' or a 'beep' is not an output device.

(b) This should have been a straightforward question but candidates often gave trivial or duplicate answers. Points which gained credit were; Can work 24/7, produces the product to a consistently high standard, do not require the ancillary facilities that human workers do, robots can work in conditions humans would deem hazardous. Points which fell short included; robots do not get bored, robots work faster than humans, robots are more efficient and many others.

(c) The answer expected was that the program the robot was using would need to be re-written or amended. Answers which described the hardware being replaced did not gain credit.

Question 6

This question proved to be demanding, despite being examined in previous years. The majority of candidates gained marks only for naming and then describing a particular scheduling algorithm. The better answers described the three possible states for any program – running, ready and blocked - and the roles played by the low-level and high-level scheduler in the management of processes.

Question 7

(a) (i) The majority of candidates correctly named this as a 1-to-many relationship and drew the correct E-R diagram.

(ii) The wording in the stem 'in third normal form' was meant to indicate to the candidate that the many-to-many relationship between TREATMENT and PATIENT must be acted on with an appropriately named link table between these entities e.g. PATIENT-TREATMENT.

(b) (i) (ii) If an explanation of either primary or foreign key is asked for the candidate should start by stating the obvious that it is an attribute. Both parts **(i)** and **(ii)** were generally well answered and supported by a correct attribute from the scenario. Some candidates did not get credit for giving attributes from a completely different problem – e.g. candidate ID - probably from the course textbook or examples from their teaching.

(iii) Practical experience with the use of a relational database would have supported the candidate well for this question where a table may be indexed on one or more attributes other than the primary key. Good answers described the secondary key being set up so that a fast search could be made on this attribute e.g. the patient's name.

Some candidates confused the term secondary key and wrongly gave a description of a candidate key.

Question 8

(i) Most candidates appreciated this was machine code or assembly language. Note a description which states *"the language understood by the processor"* gains credit whereas *"the language understood by the computer"* did not.

(ii) (iii) Again candidates who had practical experience with a language which uses OOP and a declarative language such as Prolog would have found these parts straightforward.

(iv) Surprisingly, this was the least well answered part for **Question 8**, and raised concerns about the level of practical programming experience to which candidates are exposed.

Question 9

(a) Candidates often gave tasks which would have been carried out in the previous compilation stage by the lexical analyser. Credit was given if the candidate stated this was what happened after the lexical analysis stage and the syntax analyser's input is a tokenised string. Full marks were obtained by candidates who described some of the semantic checks; the existence of labels, data type mismatch, etc.

(b) Answers to part **(ii)** were generally poor with candidates often doing little more than repeating the answer they had given for **(i)**. What was expected were two concepts; that the keywords are compared to a list of the language's acceptable keywords and secondly, that the format of each program statement is checked against the format as defined by the rules of the language - by a formal language such as a Backus Naur Form. Candidates gained credit if they illustrated this grammar checking with an example e.g. an unacceptable composition of an IF statement.

Question 10

(a) Well answered by the many candidates. Answers expected were that:

(i) the final digit must be a non-zero-digit (not a 0)
(ii) the letter W is not a terminal symbol defined by any of the rules.
(iii) a <variable-identifier> cannot end with two digits.

However, answers for **(a)** were often too superficial to score; a typical answer was *"It is not valid because it does not match with the given rules"*.

(b) This was less well answered with candidates often only stating that the '6' is a <non-zero-digit>. Candidates must reason this through either by drawing a formal parse tree or by describing all the rules which have to be used to arrive at the decision valid/invalid. For example, it was insufficient to state that the 'Ay' is a <group>. This must be reasoned by stating the intermediate steps that:

'y' is a <letter>
Hence 'y' is a <group>
So, A <group> is also a <group>

COMPUTING

Paper 9691/33
Written Paper

General

Performance across the various syllabus areas was very varied. Candidates performed well on the simulation question (**Question 5**) and the databases content (**Question 6**). Weak areas were the manipulation of binary representing a floating-point real number, the internal operation of the computer system i.e. the fetch-execute cycle, special purpose registers and their uses and the operation of various buses. Few candidates knew what was meant by a syntax diagram (**Question 10**).

Question 1

(a) (b) Many candidates mixed up their answers for **(a)** and **(b)** across the two questions, but credit was given for valid points made. Most candidates described the bootstrap program stored on ROM and described its use to load the first modules of the operating system software. Detail was often lacking for **(b)**. Only the better candidates described testing for the working of the various input-output devices or the power-up process placing the relevant address into the Program Counter register. Few candidates mentioned the relevance of the BIOS, config.sys or autoexec.bat files.

Question 2

(a) This was poorly answered. Few candidates could describe the involvement of the MDR during the fetch stage and also its use when the instruction has been decoded; e.g. for a direct addressing instruction MDR will be used to transport the contents of the operand address to the Accumulator.

(b) Few candidates were able to do more than name the three buses: data bus, address bus and control bus. Explanations were weak and often gave no more than *"the data bus is used to carry data values"* which did not score.

Question 3

(a) Sign and magnitude was well understand and most candidates scored the full three marks.

(b) Again, two's complement of integers was well understood.

(c) (i) This was a slightly different question framework to test this understanding (where the answer was given) but candidates answers were weak. The working which was expected was to isolate the mantissa and exponent, combine them together and show that this computed to 2.5.

(ii) Answered correctly by the stronger candidates. Candidates needed to realise that the mantissa was not normalised, shift the bit pattern two places to the left and adjust the exponent accordingly.

(iii) Generally poorly answered. Candidates often worked out the correct exponent +3 but could not compute the correct mantissa and hence their final answer.

Question 4

(a) All candidates scored for this question and were able to describe some key features of an Intranet.

(b) Candidates were able to state advantages of using an Intranet but disadvantages were harder to come by. This is a question where candidates should be able to apply some basic knowledge about various communication systems to the given scenario.

Question 5

(a) A well answered question with candidates drawing on a general list of the advantages of using simulation and then applying their knowledge to this scenario.

The question asked the candidates to 'describe' and the descriptions given were well considered and relevant.

(b) Candidates were resourceful with their answers and were often able to come up with five well described variables. Candidates needed to be aware with a question - part **(a)** clearly asked for *"three advantages ...",* whereas there was no indication for part **(b)** how many different variables were expected. The sensible strategy would have been to give five (on the assumption it is 'one mark for each variable stated'), and this was the approach most candidates took.

Question 6

(a) There were many points which were creditworthy. The most common answer was that a relational database will avoid data duplication. Stronger answers described the concept of program-data independence either with a description or from the practical consequence that the structure of a table can be changed without affecting existing applications programs.

(b) (i) Well understood – most candidates correctly stated the attribute GuestID.

(ii) A more demanding question which was not well understood. The candidate must first decide what is to be the primary key for the ACCOUNT table before attempting to justify a secondary key. The term 'secondary key' was confused with 'candidate key' by some candidates.

(iii) This required a description only and most candidates scored.

(iv) Again a more demanding question as there had not been a previous question part which required the candidate to state what was the relationship between GUEST AND ACCOUNT. The answer expected was that there is a one-to-many relationship between ACCOUNT and GUEST, hence the ACCOUNT table would require foreign key GuestID.

Question 7

(a) Most candidates scored but detail was often missing. Candidates appreciated that the lexical analysis stage will distinguish between the language keywords and the identifier names and that the identifier names must be checked that they follow the permitted rules and once checked, they are entered into the symbol table.

(b) (i) Well understood but sometimes let down by poorly expressed sentences; for example *"it runs faster than an intepreter"* does not make it clear that it is the final executable code which is produced which will execute faster – it could have been the execution of the compiler itself.

(ii) Again well understood. The most popular answer was that an interpreter allows for better diagnostics, and the execution halts as soon as the first error is found and the programmer is then able to immediately make the code correction.

Question 8

Paging appeared to be better understood than segmentation. Candidates could describe a program being divided into pages which are then loaded into page frames in memory of the same size. The detail then expected was that the operating system manages this with a page frame table and the major benefit of paging is that the program can be executed without the whole program loaded into memory. Pages are then swapped in and out of memory as required.

Answers for segmentation were vague and usually lacked any detail.

Question 9

(a) This should have been a straightforward two marks with only one point made for each paradigm.

(b) Candidates with a bare minimum of exposure to the use of an OOP language should have found these marks attainable. For each part what was required was an explanation of each term – class, inheritance and encapsulation – followed by its illustration by reference to the given inheritance diagram.

Question 10

(a) Generally well answered. A formal parse-tree type analysis was not expected and candidates were able to describe why each identity code was invalid with a description based on the given rules.

(b) Few candidates appreciated what was required when asked to draw a syntax diagram. Many candidates attempted to re-write the rules given. However, when a candidate had drawn the syntax diagram, it usually did score full marks and showed the key point of:

One entry and exit point only
The correct sequence for the use of the rules
The alternative paths for both the omission of more than one <digit> and the omission of any number at the end.

COMPUTING

Paper 9691/04

Project 2

Key message

Candidates should use the guidance in the syllabus 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. Candidates who prepare their work only using these text books and not the syllabus for guidance may miss out vital sections of their reports; or complete unnecessary work e.g. feasibility studies and costings.

General comments

This report provides general feedback on the overall quality of project work for GCE Advanced Level 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.

The projects submitted covered a wide variety of topics with better candidates again showing evidence of researching a problem beyond their school or college life.

In order to have the full range of marks available to the candidate, the computing project must involve a third party client whose requirements are considered and clearly documented at all stages of the system development. Centres are reminded that the project work is designed to test the candidates' understanding of the systems life cycle. The requirements are clearly set out in syllabus ***section 4***, 'The Guidance on Marking the Computing Project' ***section 7.2*** acts as a useful checklist, for teachers and candidates, setting out the expected contents of each section.

Centres are reminded that this guidance and the mark scheme have changed in 2011. Please use the up-to-date A Level Computing Syllabus for guidance on project choice, content required and how to assess candidates' project work.

Project Reports and Presentation

As usual, the presentation of most of the reports was to a very high standard, with reports word-processed and properly bound. However, candidates should ensure that only material essential to the report is included so that there is only one volume of work submitted per candidate. Candidates are reminded that only authentic letters from clients and/or users must be used to provide evidence for the Evaluation, Implementation, Investigation and Analysis sections, these letters must not be typed out by the candidates.

It is strongly recommended that the structure of the candidate's report follows that of the mark scheme set out in the syllabus. Essential evidence should not be relegated to appendices. This allows both teachers at the Centres and Moderators to easily check that work for all sections has been included. Also it is essential that the pages of the report are clearly numbered by the candidate.

From 2011 there are 3 marks available for the quality of reporting, see page 36 of the 2011 Syllabus.

Project assessment and marking

Most Centres used the marking grid on pages 48-51 of the syllabus to provide a breakdown of marks showing the marks given for each sub-section of the report. However in order to aid the process of moderation, the completed grid should include references to the appropriate pages in the candidates' reports where evidence for each section can be found. Also teachers should comment as to why they awarded the

marks for each section. Moderators have noticed that where there is a good commentary provided by a teacher the marking is usually very close to the agreed standard.

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) Quality of report.

Most candidates set out their reports in the appropriate sections and made good use of illustrations including diagrams and screenshots. Weaker candidates sometimes did not include page numbers in their reports, this meant that teachers could not clearly identify to the Moderator where evidence was to be found and those candidates were unable to cross reference items within their report.

(b) Definition Investigation and Analysis

(i) Definition - nature of the problem

Most candidates described the organisation and many identified the methods used but only the better candidates identified the origins and form of the data. This is a brief introduction for anyone who is unfamiliar with the organisation and the area under investigation.

(ii) Investigation and Analysis

In order to gain good marks candidates must clearly document client and user involvement and clearly state agreed outcomes. Candidates need to consider carefully the evidence obtained from interviews, observation of the existing system and user documents, and then ask follow up questions to fill in any gaps in the knowledge obtained about the current system or requirements for the new system. Also alternative approaches need to be discussed in depth as they would be applied to the candidate's proposed system. A detailed requirements specification should be produced based on the information collected, this must include the specific requirements of the system to be produced and not just concentrate on hardware and software.

This sub-section of the report remains the same as previous years. However Centres are reminded that a distinction has been made between the 'client', who requires the new system and the day to day 'users' of the system. In many cases the client may also be a user of the system.

(c) Design

(i) Nature of the solution

The requirements specification set out in the analysis needs to be discussed with the client and a set of measurable objectives agreed. These objectives will then form the basis for the project evaluation. Most candidates provided designs that included proposed data structures, layouts for input screens and reports required, better candidates used pseudocode and/or flowcharts to provide a detailed description of the processes to be implemented. Candidates should obtain evidence that their client has seen and commented on the design work, and then show what has changed as a result of these comments. Evidence from the solution is not required here.

(ii) Intended benefits

Candidates should describe the benefits of their intended system, not just provide a list of general statements that could apply to any system.

(iii) Limits of the scope of solution

Candidates should describe the limitations of their intended system including an estimate of the size of any files required, not just provide a list of general statements that could apply to any system.

Full marks for the design section cannot be awarded without candidates clearly supplying evidence for **(i)**, **(ii)** and **(iii)**.

(d) Software Development, Programming Testing and Installation

(i) Development

Evidence of development should include program listings of code written by the candidate, data structures used and evidence of tailoring of software packages. This should match the design specification in **(c)(i)**.

(ii) Programming

It is important that the programming assessed in this subsection is written by the candidate and not produced as a result of tailoring part of a software package. Candidates need to show that they can apply the programming skills developed at AS Level in Paper 2 to a real situation. This includes technical competence and ensuring that their program could be maintained by writing self-documented code.

(iii) 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/boundary cases, together with expected results for all tests. For top marks to be awarded the test plan should clearly identify that all parts of the system have been tested. Yet again, many candidates only tested the validation and navigation aspects of their system, and omitted to test that the system did what it is supposed to do, thus being unable to gain marks in the highest band for this section.

(iv) Installation

It was pleasing to see more candidates providing a detailed implementation plan that contained details of user testing, user training and system changeover. However for good marks to be awarded there should be evidence to show that the system has been seen and used and these plans have been agreed with the client.

(iii) Documentation

(i) Systems Maintenance Documentation

The standard of work provided for this section is usually high. This sub-section of the report is now a systems maintenance document. Please see page 42 of the 2011 Syllabus for details. For top marks to be awarded the candidate should explain any adaptive maintenance that could be undertaken.

(ii) User Documentation

This section was completed to a good standard by most candidates. Centres are again reminded that for full marks the candidate must include an index and a glossary, and the guide needs to be complete including details of how to install the new system, backup routines and a guide to common errors. Also good on-screen help should exist where this is a sensible option.

(iv) Evaluation

Centres are again reminded in order to gain high marks candidates need to provide a detailed evaluation that included the content set out in the guidance for marking projects section of the syllabus. Many candidates provided scant evidence for this section, if this is the case then there are few marks that can be awarded.

Centres are reminded that possible extensions and the good and bad points of their final system are not now required for the report.

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

Candidates need to consider each objective set and explain how their project work met the objective or explain why the objective was not met. Candidates should also indicate where the evidence, probably from testing or feedback from the users of the system, could be found in their report to support these conclusions.

(ii) Evaluate the client's and users' response to the system

Again Centres are reminded that this response needs to be clearly provided from the client and user(s) showing that they have used the system, not just reported by the candidate. The candidate should then evaluate their client's and users' responses.

For evidence in this section to be creditworthy, the candidate must include original letters, preferably on headed notepaper, signed by the client and not typed and/or composed by the candidate.