COMPUTING

Paper 9691/11
Written Paper

General comments

The standard of candidates' work was broadly similar to previous years.

It is evident, however, that a significant number of candidates are learning certain topics by rote, without necessarily understanding the topics. Questions which required an application of knowledge were overall less well answered. 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 may be challenging so Centres and candidates, will need to prepare more thoroughly.

One final note regards the exam papers themselves; 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) Candidates need to develop their understanding of software. It was common to see that software was stated as programs or set of instructions with no reference to carrying out any tasks. Candidates also needed to improve on their answers to part **(ii)**, as few correct answers were seen. Operating software was often quoted as simply to control the hardware. For part **(iii)**, candidates needed to write more precise answers when referring to application software.

(b) This question was reasonably well answered. In particular, there were some very good attempts at describing the use of web browsers.

(c) Candidates need to develop their understanding of the term "data logging". It was common to see answers such as "to log data" which was really just rewording the stem of the question. The most common mistake was "to take readings every TWO hours" which was actually the TOTAL length of the experiment. However, many of the better candidates gained the full 3 marks.

Question 2

The majority of candidates could list that a knowledge base, inference engine and a Human-Computer-Interface are the parts that, with the given rule base, make up the parts of an expert system. However, candidates need to develop their understanding of what these parts are used for. Candidates need to be aware of the difference between information and data. A knowledge base contains facts, rather than just data. An inference engine applies the rules in the rule base to the facts in the knowledge base. Unfortunately, several candidates also ignored the stem of the question and lost marks by including rule base in their discussion.

Question 3

The majority of candidates listed appropriate applications where sound and animation would be used, but acceptable justifications were much rarer.

Question 4

The majority of candidates could describe that a hardware driver would allow communication between the hardware and the operating system, but many candidates just described how a hard disk drive worked.

The second question about virus checkers gained more marks for the candidates. However very few, if any, mentioned that it worked continuously and even fewer mentioned that it worked in the background. It was common to gain marks for mentioning the deletion of viruses or keeping the system virus-free. Only the better candidates mentioned the idea of quarantine.

Question 5

This question was well answered by the majority of candidates. However, some candidates needed to understand better what the question required, as they listed stages of the system life cycle or went into detail of how the drilling machine might be used.

Question 6

(a) What the control unit does is to control the computer or processor. Candidates needed better understanding in order to answer this question. Many candidates seemed to think the control unit fetched a program rather than an instruction. Very few mentioned that the control unit sends control signals to other parts of the processor; however, many candidates did indicate that it "tells" the processor to execute instruction rather than send signals.

(b) In part **(i)**, the idea of a buffer was very well known, although weaker candidates left out the important idea that it held its data temporarily. Part **(ii)** showed that candidates needed to improve on their understanding of the concept of an interrupt. Several candidates did not mention the role of the processor which is equally vital to the answer. Some candidates also incorrectly stated that an interrupt was a signal to the user.

(c) The better candidates gave good descriptions of how the buffer and interrupts are used to transfer data from a hard disk to primary memory. Candidates need to understand that a buffer and primary memory are not able to perform any actions themselves, they do not themselves transfer data nor do they generate an interrupt. The processor transfers data into the buffer from the disk drive. When the buffer is full an interrupt is sent from the disk drive to the processor so that the buffer contents can be emptied into primary memory. Candidates also need to understand that interrupts are part of the normal functioning of a computer system and not just generated because an error has occurred.

Question 7

(a) The use of a hashing algorithm was very well answered – not unusual to see full marks and very often succinctly worded. However, some candidates would benefit from further developing their understanding of hashing and preventing data corruption.

(b) In part **(i)**, the majority of candidates had some idea of what a collision was; although in many cases marks were lost by adding something that contradicted their previous statement. A number of candidates seemed to think collisions stopped damaging a file that had random access. Part **(ii)** was slightly less well answered. Describing the two methods to overcome clashes produced some interesting wrong answers like change the file names, alter the hashing algorithm or have a second hash algorithm. One or two candidates even mentioned circuit and packet switching. Common correct answers were use an overflow area with a flag/tag to show it is in use or to search serially (sometimes sequentially) for the next free memory slot address. Buckets were also mentioned quite often.

Question 8

(a) In part **(i)**, the expected input device for an airport terminal building was a touch screen. However, a large number of candidates wrongly suggested a keyboard, a mouse or a barcode reader. The justification given for choice of device was often to avoid vandalism, but many did correctly suggest that a touch screen afforded a simple to use interface. Very few candidates mentioned the other correct advantage of a touch screen: a limited choice of options. However, a significant number did correctly state that a touch screen could be used as both an input and an output device. Part **(ii)** gave a similar standard of answers. The only acceptable devices here were screen or speaker. Devices such as printers or VDU were not accepted in this application.

(b) Candidates needed to relate their answers to the given scenario. The answers given here were often imprecise. There was strong evidence of producing answers based on previous mark schemes without really considering whether or not they were relevant in this context.

Question 9

(a) In part **(i)**, serial simplex generally was very well understood and clearly answered. A substantial number of candidates thought serial transmission was just one direction of transmission and then stated the same for simplex.

In part **(ii)**, candidates needed to improve their understanding of parallel, full duplex transmission. Parallel was frequently described as data travelling in both directions and full duplex being data transmitted in one direction at a time.

(b) The idea of what a protocol is was generally well understood by the candidates; but what it is used for, i.e. to control/govern data communication, was less well known. Candidates often stated that it was just to allow communication which was not regarded as enough to gain a mark.

(c) Parity being used to detect errors was generally well known but often weakly answered. Relatively few candidates mentioned that each byte has a bit reserved as a parity bit. Most candidates did mention that the parity could be odd or even but did not fully understand the idea that the parity bit is what makes the numbers of 1's become odd or even. The idea that parity detects errors after transmission was stated but in reference to the sender; it is the receiver that does the parity checking to detect errors in transmission. Some candidates needed to further develop their understanding of parity.

Question 10

(a) The vast majority of candidates correctly completed the truth table and realised that the two given logic gates could be replaced by a NAND gate.

(b) The majority of candidates correctly completed the first part of the truth table. Only a minority of candidates knew how to complete the remaining input values for A, B and C and the output values for D, E and F. Candidates need to understand that input values for a truth table should 'count up' in binary from all inputs as 0 until all inputs are 1.

COMPUTING

Paper 9691/12
Written Paper

General comments

The standard of candidates' work was broadly similar to previous years.

It is evident, however, that a significant number of candidates are learning certain topics by rote, without necessarily understanding the topics. Questions which required an application of knowledge were overall less well answered. 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 may be challenging so Centres and candidates, will need to prepare more thoroughly.

One final note regards the exam papers themselves; 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) In part **(i)**, the majority of candidates correctly defined hardware as the physical/electronic components of a computer. However, a significant number of candidates gained no credit for describing hardware as only peripherals or making imprecise comments about things that can be touched. In part **(ii)**, most candidates gained credit for being able to state that an input device is a peripheral, but very few candidates could accurately define an input device and why it is needed. Candidates need to remember that restating the words of the question gains no credit.

(b) Candidates need to improve on their responses to this question. A large majority of candidates did not appear to know what a desktop publisher (DTP) is used for and many confused it with presentation software and made references to using multimedia.

(c) How mark recognition works when used to mark multiple choice examination papers did not seem to be well understood. The majority of answers revolved around how a candidate would fill in such an exam paper rather than the operational detail of an optical mark reader.

Question 2

The majority of candidates could list that a knowledge base, a rule base and a Human-Computer-Interface are the parts that, with the given inference engine, make up the parts of an expert system. However, candidates need to develop their understanding of what these parts are used for. Candidates need to be aware of the difference between information and data. A knowledge base contains facts, rather than just data. A rule base contains the rules that the inference engine applies to the facts in the knowledge base. Unfortunately, several candidates also ignored the stem of the question and lost marks by including inference engine in their discussion.

Question 3

The majority of candidates listed appropriate applications where graphs and hard copy reports would be used, but acceptable justifications were much rarer.

Question 4

The majority of candidates could describe that a disk formatter would erase the contents of the whole disk, but some candidates seemed to be under the mistaken impression that only selective files could be erased in order to make more space for new files. Only the better candidates offered further detail: that formatting divides a disk surface into tracks and sectors and that this is done in order to prepare a new disk for use. Very few candidates mentioned FAT initialisation.

Most candidates knew that file compression reduces the size of files and why this would be used. Very few candidates could explain how compression is possible without losing information. A few candidates could state that compression can group several files into one file and that decompression is required to restore the original files.

Question 5

This question was well answered by the majority of candidates. However, some candidates needed to understand better what the question required, as they listed stages of the system life cycle or went into detail of how the drilling machine might be used.

Question 6

(a) The functions of the ALU did not seem to be well understood. Many candidates ascribed a much higher level of functionality to this part of the processor. Candidates need to understand that the ALU is a series of logic gates that are set by the opcode (contents of the current instruction register) to perform an arithmetic operation (such as add, subtract) or a logical operation (such as comparing) using operands supplied to the ALU from registers or the accumulator.

(b) In part **(i)**, the majority of candidates knew that a buffer is a temporary storage area. However, many candidates seemed to think that a buffer is a device, when in fact it is a group of memory locations.

In part **(ii)**, candidates need to understand that an interrupt is merely a signal that is generated by hardware (or software) and sent to the processor to get processor time.

(c) The better candidates gave good descriptions of how the buffer and interrupts are used to transfer data from primary memory to a printer. Candidates need to understand that a buffer and primary memory are not able to perform any actions themselves, they do not themselves transfer data nor do they generate an interrupt. The processor transfers data into the buffer from primary memory. The printer then fetches the data from the buffer and generates an interrupt when the buffer is empty. Candidates also need to understand that interrupts are part of the normal functioning of a computer system and not just generated because an error has occurred.

Question 7

(a) Most candidates knew that serial files have records in chronological order (i.e. in order of arrival). Fewer candidates were able to state that sequential files have records arranged in key field order. Some candidates need to improve their understanding of files and record.

(b) In part **(i)**, most candidates stated that a new record would be added to the end of the serial file. A small number of candidates were under the wrong impression that a record would be added to the top of the file.

In part **(ii)**, most candidates needed to understand better how a record would be added to a sequential file. A few candidates incorrectly described indexing. Many candidates wrongly thought that the record could be added to the end and then all records could be sorted. For this to be possible, records would have to be read into an array and then saved back to a file.

The standard method candidates should be aware of is that the sequential file is read, a record at a time, starting at the beginning. The key field of each record is compared to the key field of the new record. If the new key is greater, the existing record is written to a new file. If the new key is smaller, the new record is written to the new file. Once the new record is written the remainder of the old file is copied to the new file.

Question 8

(a) Most candidates stated suitable input and output devices for a pocket sized computer game. Only the better answers gave a good justification.

(b) Candidates needed to relate their answers to the given scenario. The majority of answers made little reference to what would be a suitable layout for this computer game.

Question 9

(a) The majority of candidates were able to describe what each of the terms meant: serial, parallel, simplex and half-duplex. There were a significant number of candidates, however, who confused these terms or whose descriptions were so imprecise as to not gain credit.

(b) Most candidates knew that a protocol is a set of rules. Fewer candidates could go on to say that these govern the transmission of data.

(c) A large number of candidates explained echoing very well. However, some candidates would benefit from developing their understanding of echoing.

Question 10

(a) The vast majority of candidates correctly completed the truth table and realised that the two given logic gates could be replaced by a NOR gate.

(b) The majority of candidates correctly completed the first part of the truth table. Only a minority of candidates knew how to complete the remaining input values for A, B and C and the output values for D, E and F. Candidates need to understand that input values for a truth table should 'count up' in binary from all inputs as 0 until all inputs are 1.

COMPUTING

Paper 9691/13
Written Paper

General comments

The standard of candidates' work was broadly similar to previous years.

It is evident, however, that a significant number of candidates are learning certain topics by rote, without necessarily understanding the topics. Questions which required an application of knowledge were overall less well answered. 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 may be challenging so Centres and candidates, will need to prepare more thoroughly.

One final note regards the exam papers themselves; 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) In part **(i)**, the majority of candidates correctly defined hardware as the physical/electronic components of a computer. However, a significant number of candidates gained no credit for describing hardware as only peripherals or making imprecise comments about things that can be touched. In part **(ii)**, most candidates gained credit for being able to state that an output device is a peripheral, but very few candidates could accurately define an output device and why it is needed. Candidates need to remember that restating the words of the question gains no credit.

(b) This question was reasonably well answered. There were very few issues raised by the answers offered by the candidates.

(c) The majority of candidates knew what a barcode was and that it consisted of parallel black and white bars of varying thicknesses. They were also generally aware that the bars were read using a laser light source. However, candidates needed to improve on their knowledge of barcodes as very few seemed to be aware of the significance of the line thickness and how this information was converted into a form that the computer could understand. Candidates also need to develop their understanding of how barcodes are used at checkouts to locate product information and how they link into automatic stock control systems.

Question 2

The majority of candidates could list that a knowledge base, inference engine and a rule base are the parts that, with the given human computer interface (HCI), make up the parts of an expert system. However, candidates need to develop their understanding of what these parts are used for. Candidates need to be aware of the difference between information and data. A knowledge base contains facts, rather than just data. An inference engine applies the rules in the rule base to the facts in the knowledge base and a rule base contains all the rules applied to data in the knowledge base. Unfortunately, several candidates also ignored the stem of the question and lost marks by including HCI in their discussion.

Question 3

The majority of candidates listed appropriate applications where images and interactive presentations would be used, but acceptable justifications were much rarer.

Question 4

In part **(i)**, the majority of candidates could describe that a hardware driver would allow communication between the hardware and the operating system, but many candidates just described how a hard disk drive worked.

In the second part, the majority of candidates could describe that a disk formatter would erase the contents of the whole disk, but some candidates seemed to be under the mistaken impression that only selective files could be erased in order to make more space for new files. Only the better candidates offered further detail: that formatting divides a disk surface into tracks and sectors and that this is done in order to prepare a new disk for use. Very few candidates mentioned FAT initialisation.

Question 5

This was reasonably well answered with the majority gaining 2 or 3 marks for answers such as: *is it economically feasible, is it technically feasible* and *is the final solution legal.* However, it was much rarer to see the following answers: *will the computerisation cause unemployment, can it be produced in a reasonable time frame* and *are the social effects likely to be too damaging.*

Question 6

(a) Several candidates gained one mark here for stores part of the program currently in use. However, very few gained any more marks for mentioning that it stores the part of the OS currently in use or that it stores the results of processing.

(b) In part **(i)**, the majority of candidates knew that a buffer is a temporary storage area. However, many candidates seemed to think that a buffer is a device, when in fact it is a group of memory locations.

In part **(ii)**, candidates need to understand that an interrupt is merely a signal that is generated by hardware (or software) and sent to the processor to get processor time.

(c) The better candidates gave good descriptions of how the buffer and interrupts are used to transfer data from primary memory to the hard disk. Candidates need to understand that a buffer and primary memory are not able to perform any actions themselves, they do not themselves transfer data nor do they generate an interrupt. The processor transfers data into the buffer from primary memory. The hard disk then fetches the data from the buffer and generates an interrupt when the buffer is empty. Candidates also need to understand that interrupts are part of the normal functioning of a computer system and not just generated because an error has occurred.

Question 7

(a) This question was reasonably well answered. However, several candidates suggested that the records were stored in alphabetical order effectively ignoring the stem of the question.

(b) In part **(i)**, candidates needed to demonstrate a better understanding of the question. Many candidates produced very unclear diagrams. Part **(ii)** was still weak, although the first level index was often shown correctly.

Question 8

(a) In part **(i)**, the expected input device for air traffic control was a touch screen or trackerball. However, a large number of candidates wrongly suggested a keyboard or a microphone. The justification given for the choice of device was often to type in numbers of aeroplanes or to speak to the pilots. Very few candidates mentioned the real advantage of a touch screen: it allowed quick selection of a particular flight by simply touching the screen. However, a significant number did correctly state that a touch screen could be used as both an input and an output device. Part **(ii)** gave a similar standard of answers. The only acceptable devices here were screen or speaker. Devices such as printers or VDU were not accepted in this application.

(b) Candidates needed to relate their answers to the given scenario. There was strong evidence of producing answers based on previous mark schemes without really considering whether or not they were relevant in this context.

Question 9

(a) Often only half of the definition was given and candidates described the type of transmission but omitted the definition of serial or parallel.

(b) The idea of what a protocol is was generally well understood by the candidates; but what it is used for, i.e. to control/govern data communication, was less well known. Candidates often stated that it was just to allow communication which was not regarded as enough to gain a mark.

(c) This was generally reasonably well answered but several candidates described the use of parity and how odd and even parity was used to do error checking. However, many candidates gained two marks for some idea of the value of the bytes in the block being found and that this was e-checked at the receiving end as a check for transmission error.

Question 10

(a) The vast majority of candidates correctly completed the truth table and realised that the two given logic gates could be replaced by a NAND gate.

(b) The majority of candidates correctly completed the first part of the truth table. Only a minority of candidates knew how to complete the remaining input values for A, B and C and the output values for D, E and F. Candidates need to understand that input values for a truth table should ‘count up’ in binary from all inputs as 0 until all inputs are 1.

COMPUTING

Paper 9691/21
Written Paper

Key message

For this component candidates need to have practical programming experience in the programming facilities of a chosen high-level language (such as Basic, Visual Basic or Pascal). Candidates need to write their own programs, adapt programs written by others, and write programs from pseudocode prepared by others.

Question 1

Parts **(a)** and **(b)**

Most candidates were able to complete the structure diagram. A small number of candidates needed to improve on the format of their structure charts as they added subtasks vertically rather than horizontally.

In part **(c)** the reasons given for dividing the main task into smaller tasks were creditworthy. A few candidates claimed that smaller tasks were easy to understand and debug. This is not necessarily the case. Smaller tasks are **easier** to understand and debug than a large, main task.

In part **(d)** the majority of candidates were able to explain that variables declared within a module were local to that module and a variable with the same identifier declared within another module would be stored in a different memory location.

Part **(e)** was well answered by most candidates. A small minority of candidates did not fully understand that the Boolean operator OR was required as an answer.

Part **(f)** showed that the majority of candidates had done some practical programming in a high-level programming language, although the language stated and the answer provided did not always match. The best candidates gave almost completely correct answers. This component requires candidates to practise programming in a practical way and experience all the programming language facilities mentioned in the syllabus, including use of built-in functions for string manipulation. Less able candidates did not seem to have enough experience in separating out substrings from a string or use the built-in function for finding the length of a string. In part **(f) (ii)** the better candidates were able to explain why BIKE46 was not a valid bike registration and also give reference to a line of their program code where this bike registration was found to be invalid. Many candidates listed the fact that lower case letters were present in the given registration, but a valid registration should only have upper case letters. However, their program code did not check for this.

Part **(g)** resulted in many imprecise answers.

(i) Many candidates needed to improve on their answers as they chose the wrong type of testing; white box testing was the correct answer.

(ii) Some candidates needed to understand that alpha testing is not done by the programmer who wrote the program, but by other employees of the software house, usually staff who are specifically employed as software testers. Alpha testing would be performed with software that was essentially finished, apart from correcting errors still found at this testing stage.

Question 2

The majority of candidates correctly completed the trace table. Common errors were to miss out the value 4 for position, before the row value increases to 2 and position is reset to 1. A significant minority of candidates did not fully understand how to complete the heading for the BikePlace array correctly. This was often the case even though the values in the Row and Position column were given correctly.

Part **(b) (i)** was done well by the better candidates. The majority of candidates seemed to find this question difficult. Candidates need to have lots of practice at converting algorithms and flowcharts into code using a high-level programming language and getting their program to work correctly. Many answers showed weaker understanding of how to implement correctly the nested loops (which are clearly indicated in the flowchart) using FOR NEXT loops, even though the candidates had just performed a dry run of this flowchart and should have some understanding of its function. The pseudocode notation used in the flowchart was often adopted for the programming code. However, the chosen programming language syntax was at times not followed. For example, assignment in a programming language does not use the ← symbol used in pseudocode. Visual Basic does not use [ ] for array indices.

The better candidates answered part **(b) (ii)** well. The majority of candidates need to understand that a reserved word is any word in the vocabulary of a programming language which can only have the meaning defined in that language. Many candidates could give valid examples from their program code in part **(b) (i)**. However, some candidates gave function identifiers as reserved words. Candidates need to understand that functions can be redefined and as such their identifiers are not classed as reserved words.

Part **(c) (i)** was often answered correctly, although occasionally a non-zero answer was seen. The correct answer to part **(ii)** is runtime error. Most candidates gave arithmetic or logic error as an answer, which was given credit. A minority of candidates needed to improve on their answer as they gave syntax error as an answer. Part **(iii)** was rarely answered correctly. Many candidates seemed to think that the formula could be changed. The better candidates realised that the value in brackets had to be tested before the division takes place and if it was zero the program would have to output a message.

Answers to part **(d)** showed that few candidates seemed to have practical experience of using debugging tools listed in the syllabus. Most answers were imprecise. Candidates should have practical experience of debugging by setting a break point at the beginning of their program code under scrutiny so that their program runs normally until the break point is reached. Then they can check the current value of variables using a variable check (variable watch).

Question 3

The majority of candidates produced good report designs. To gain full credit candidates had to provide a suitable report title (not just Report), include the company name (Super Bikes) and the month that the report referred to. To show the required information (5 bikes, how many times they were hired, how much income each generated, and what repairs each required) a table was the best layout. The report needed to make use of the whole frame given in the answer space. Some candidates needed to improve on their answer as the question required them to design a report layout and they had instead given a screen design, which could not gain full marks.

Question 4

The majority of candidates answered parts **(a)** and **(b)** very well. Although both BikeID and BikeType are of the same data type, many candidates did not fully understand this, using the term String or Text for one and Alphanumeric for the other. Part **(c)** required candidates to demonstrate their knowledge of programming. The better candidates clearly had used record types in their chosen programming language and correctly gave the heading of a record definition. The majority of candidates, however, had weaker knowledge of this topic, usually getting some credit for declaring the fields with the correct data type.

In part **(d) (i)** the majority of candidates gained credit for mentioning that a function always returns a value. The better candidates understood that a subroutine to add a record to a file would not require any values to be returned to the calling program. Some candidates did not fully understand and mistakenly thought that the fields of the record would be required to be passed back to the calling program.

In part **(d) (ii)** the better candidates gave answers that gained credit. Others needed to understand the difference in parameter passing. Passing a parameter by value means a local copy of the data is used within the procedure. Passing a parameter by reference means the address of the memory location of the data to be used is passed to the procedure.

In part **(d) (iii)** the best candidates realised that the parameters that would need to be required to be passed to the procedure would be filename and bike record. Either of these would have earned the candidate full credit.

COMPUTING

Paper 9691/22
Written Paper

Key message

For this component candidates need to have practical programming experience in the programming facilities of a chosen high-level language (such as Basic, Visual Basic or Pascal). Candidates need to write their own programs, adapt programs written by others, and write programs from pseudocode prepared by others.

Question 1

Parts **(a)** and **(c)**

Most candidates were able to complete the structure diagram. A small number of candidates needed to improve on the format of their structure charts as they added subtasks vertically rather than horizontally.

In part **(b)** the reasons given for dividing the main task into smaller tasks were creditworthy. A few candidates claimed that smaller tasks were easy to understand and debug. This is not necessarily the case. Smaller tasks are **easier** to understand and debug than a large, main task.

In Part **(d)**, candidates needed to be more precise. The output for the input "green" was usually given correctly as Invalid. However, few candidates realised that the input "Black" was invalid as it was not equal to "black". The input "grey" caused many incorrect answers. Candidates need to understand that the pseudocode statement OUTPUT Colour " is valid" will produce the output: grey is valid and not just valid.

Part **(e)** showed that the majority of candidates had done some practical programming in a high-level programming language, although the language stated and the answer provided did not always match. The best candidates gave almost completely correct answers. This component requires candidates to practise programming in a practical way and experience all the programming language facilities mentioned in the syllabus, including use of built-in functions for string manipulation. Less able candidates did not seem to have enough experience in separating out substrings from a string or use the built-in function for finding the length of a string. In part **(e) (ii)** the better candidates were able to explain why 487HIRE was not a valid car registration and also give reference to a line of their program code where this car registration was found to be invalid. Many candidates listed the fact that there were three digits in the given registration, but a valid registration should only have two digits. However, their program code did not check for this. A more apparent reason for showing that the registration was invalid was that it would not pass the length check. Often the output statement was given as the piece of code that detected this.

Part **(f)** resulted in many imprecise answers.

(i) Candidates need to understand that alpha testing is not done by the programmer who wrote the program, but by other employees of the software house, usually staff who are specifically employed as software testers. Alpha testing would be performed with software that was essentially finished, apart from correcting errors still found at this testing stage.

(ii) Candidates need to understand that beta testing is performed by people outside the software house. This may be potential users or specific customers. Beta testing is performed after alpha testing and its purpose is to get feedback from these potential users and to find errors missed during in-house testing.

Question 2

The majority of candidates correctly completed the trace table. Common errors were to miss out the value 5 for position, before the row value increases to 2 and position is reset to 1. A significant minority of candidates did not fully understand how to complete the heading for the ParkingSpace array correctly. This was often the case even though the values in the Row and Position column were given correctly. Some candidates needed to have used the input values that were given in the stem of the question.

Part **(b)** was done well by the better candidates. The majority of candidates seemed to find this question difficult. Candidates need to have lots of practice at converting algorithms and flowcharts into code using a high-level programming language and getting their program to work correctly. Many answers showed weaker understanding of how to implement correctly the nested loops (which are clearly indicated in the flowchart) using WHILE loops, even though the candidates had just performed a dry run of this flowchart and should have some understanding of its function. The pseudocode notation used in the flowchart was often adopted for the programming code. However, the chosen programming language syntax was at times not followed. For example, assignment in a programming language does not use the ← symbol used in pseudocode. Visual Basic does not use [ ] for array indices.

Part **(c) (i)** was usually answered correctly, although occasionally a non-zero answer was seen. The correct answer to part **(ii)** is runtime error. Most candidates gave arithmetic or logic error as an answer, which was given credit. A minority of candidates needed to improve on their answer as they gave syntax error as an answer. Part **(iii)** was rarely answered correctly. Most candidates seemed to think that the formula could be changed. The better candidates realised that the value in brackets had to be tested before the division takes place and if it was zero the program would have to output a message.

Answers to part **(d)** showed that few candidates seemed to have practical experience of using debugging tools listed in the syllabus. Most answers were imprecise. Candidates should have practical experience of debugging by setting a break point at the beginning of their program code under scrutiny so that their program runs normally until the break point is reached. Then they can check the current value of variables and use stepping to check the value of variables as they execute one instruction at a time.

Question 3

The majority of candidates answered parts **(a)** and **(b)** very well. Although both CarReg and Make are of the same data type, many candidates did not fully understand this, using the term String or Text for one and Alphanumeric for the other. Part **(c)** required candidates to demonstrate their knowledge of programming. The better candidates clearly had used record types in their chosen programming language and correctly gave the heading of a record definition. The majority of candidates, however, had weaker knowledge of this topic, usually getting some credit for declaring the fields with the correct data type. The better candidates were able to write a procedure to add a record to the end of a file. Few candidates managed to give a correct procedure heading. Some candidates needed to have a better understanding of file handling. This is another topic candidates need to practise by writing and running their own programs. In part **(d)** only the better candidates gave answers that gained credit. Candidates need to understand the difference in parameter passing. Passing a parameter by value means a local copy of the data is used within the procedure, leaving the variable in the main program unaffected. Passing a parameter by reference means the address of the memory location of the data to be used is passed to the procedure. This means that any changes to the content of this memory location will also be reflected in the main program from where the procedure was called.

Question 4

The majority of candidates produced good report designs. To gain full credit candidates had to provide a suitable report title (not just Report), include the company name (Super Cars) and the month that the report referred to. To show the required information (10 cars, how many times they were hired and how much income each generated) a table was the best layout. The report needed to make use of the whole frame given in the answer space. Some candidates needed to improve on their answer as the question required them to design a report layout and they had instead given a screen design, which could not gain full marks.

COMPUTING

Paper 9691/23
Written Paper

Key message

For this component candidates need to have practical programming experience in the programming facilities of a chosen high-level language (such as Basic, Visual Basic or Pascal). Candidates need to write their own programs, adapt programs written by others, and write programs from pseudocode prepared by others.

Question 1

Parts **(a)** and **(c)**

Most candidates were able to complete the structure diagram. A small number of candidates needed to improve on the format of their structure charts as they added subtasks vertically rather than horizontally.

In part **(b)** the reasons given for dividing the main task into smaller tasks were creditworthy. A few candidates claimed that smaller tasks were easy to understand and debug. This is not necessarily the case. Smaller tasks are **easier** to understand and debug than a large, main task.

In Part **(c)** candidates needed to be more precise. The output for the input "scrambler" was usually given correctly as Invalid. However, few candidates realised that the input "trail" was invalid because of the logic in the statement. The input "road" caused many incorrect answers. Candidates need to understand that the pseudocode statement OUTPUT BikeType " is valid" will produce the output: road is valid and not just valid.

Part **(d)** only needed a change of AND to OR and most candidates recognised this.

Part **(e)** showed that the majority of candidates had done some practical programming in a high-level programming language, although the language stated and the answer provided did not always match. The best candidates gave almost completely correct answers. This component requires candidates to practise programming in a practical way and experience all the programming language facilities mentioned in the syllabus, including use of built-in functions for string manipulation. Less able candidates did not seem to have enough experience in separating out substrings from a string or use the built-in function for finding the length of a string. In part **(e) (ii)** the better candidates were able to explain why 487BK was not a valid bike registration and also give reference to a line of their program code where this bike registration was found to be invalid. Many candidates listed the fact that the digits and the letters were the wrong way round in the given registration. However, their program code did not check for this, or anything similar. Often the output statement was given as the piece of code that detected this.

Part **(f)** resulted in many imprecise answers.

(i) Candidates are familiar with the term 'black box', but were unable to state what it means.

(ii) Similarly 'white box testing' appears familiar, but many candidates did not fully understand what it means.

Question 2

The majority of candidates correctly completed the trace table. Common errors were to miss out the value 5 for position, before the row value increases to 2 and position is reset to 1. A significant minority of candidates did not fully understand how to complete the heading for the BikeSpace array correctly. This was often the case even though the values in the Row and Position column were given correctly. Some candidates needed to have used the input values that were given in the stem of the question.

Part **(b)** was done well by the better candidates. The majority of candidates seemed to find this question difficult. Candidates need to have lots of practice at converting algorithms and flowcharts into code using a high-level programming language and getting their program to work correctly. Many answers showed weaker understanding of how to implement correctly the nested loops (which are clearly indicated in the flowchart) using REPEAT…UNTIL loops, even though the candidates had just performed a dry run of this flowchart and should have some understanding of its function. The pseudocode notation used in the flowchart was often adopted for the programming code. However, the chosen programming language syntax was at times not followed. For example, assignment in a programming language does not use the ← symbol used in pseudocode. Visual Basic does not use [ ] for array indices.

Part **(c) (i)** was usually answered correctly, although occasionally a non-zero answer was seen. The correct answer to part **(ii)** is runtime error. Most candidates gave arithmetic or logic error as an answer, which was given credit. A minority of candidates needed to improve on their answer as they gave syntax error as an answer. Part **(iii)** was rarely answered correctly. Most candidates seemed to think that the formula could be changed. The better candidates realised that the value in brackets had to be tested before the division takes place and if it was zero the program would have to output a message.

Answers to part **(d)** showed that few candidates seemed to have practical experience of using debugging tools listed in the syllabus. Most answers were imprecise. Candidates should have practical experience of debugging by setting a break point at the beginning of their program code under scrutiny so that their program runs normally until the break point is reached. Then they can check the current value of variables and use stepping to check the value of variables as they execute one instruction at a time.

Question 3

The majority of candidates produced good report designs. To gain full credit candidates had to provide a suitable report title (not just Report), include the company name (Super Bikes) and the month that the report referred to. To show the required information (3 insurance groups, all the bikes hired and how much income each generated) a table was the best layout. The report needed to make use of the whole frame given in the answer space. Some candidates needed to improve on their answer as the question required them to design a report layout and they had instead given a screen design, which could not gain full marks.

Question 4

The majority of candidates answered parts **(a)** and **(b)** very well. Part **(c) (i)** and **(ii)** required candidates to demonstrate their knowledge of programming. The better candidates clearly had used record types in their chosen programming language and correctly gave the heading of a record definition. The majority of candidates, however, had weaker knowledge of this topic, usually getting some credit for declaring the fields with the correct data type. The better candidates were able to write code that manipulated a file. Some candidates needed to have a better understanding of file handling. This is another topic candidates need to practise by writing and running their own programs

COMPUTING

Paper 9691/31
Written Paper

General

The performance of the candidates on this paper was very varied. Candidates are often proficient at re-producing bookwork definitions which are usually the lead-in to a question - examples would be **Questions 3(a)** and **5(a)**. However, candidates then are less confident in applying the knowledge about an area of the syllabus to some practical application - e.g. **Question 5(c)** – or to a skills-based task – e.g. **Question 5(b)**.

It is appreciated that English is not the first language for the majority of candidates who sit this paper. Examiners make every effort to take this into account, however, some questions do require a clear explanation to convey the candidate's understanding – e.g. **Question 3(b)** and **9(b)(ii)**.

Question 1

Candidates needed to improve on their understanding of foreign keys for part **(a)(i)**. They often scored only one of the available two marks by not stating that a foreign key is an attribute.

For part **(a)(ii)** a concise statement such as *"The primary key in table X links to a foreign key attributes in Table Y"* would have secured the full 3 marks.

For part **(b)(i)** candidates had to be resourceful and add appropriate attributes to the Customer and Product tables and found no difficulty with this. For part **(b)(ii)** the most common answer seen did correctly have the ProductID and CustomerID present in the Order table. However, for the second mark candidates had to work out from the rubric that the table would have a composite primary key of CustomerID + OrderDate and this was rarely seen.

The final part **(c)** required good communication of the candidate answer and often candidates gave an example from the scenario for part **(b)** – typically, that there would be no need for the customer name to be present in the Order table. For the second mark some mention of the need for normalised tables was expected.

Question 2

Part **(a)** saw strong responses, however few candidates were able to compute the real number for part **(b)(ii)**. Candidates were able to explain the reason for a normalised representation in part **(c)(ii)** but needed further understanding to transform the given bit pattern to its normalised representation.

Question 3

For part **(a)** there were many points which could have been made and the majority of candidates scored the full two marks.

Part **(b)(i)** was well answered. The most concise answer would have been that "the contents of the program Counter are incremented". The most popular candidate answer stated "The Program Counter is incremented by one" which gained credit.

Candidates needed to improve on their answers to part **(ii)** as they rarely scored a mark with incorrect statements such as *"the contents of the Memory Address Register are copied to the Memory Data Register".* The meaning is best expressed with two sentences. *"The Memory Address Register holds the address to be used. The data value at this address in MAR is copied to the Memory Data Register".*

Part **(b)(iii)** required only mention of the instruction in CIR being decoded and then executed. Mention of further access to memory - e.g. for a jump instruction - was not expected.

For part **(c)** the quality of the answers was varied. Candidates who followed the suggestion of a diagram to illustrate their answer generally faired better than those who relied on a description only. Despite being highlighted as a weakness in a previous Principle Examiner report, indexed addressing is not well understood.

Question 4

Candidates generally scored well on this question. The process of interpretation was well understood although some candidates incorrectly stated that object code is produced by an interpreter. Candidates were usually able to give two advantages of using a compiler. The most popular answers were that the object code produced will execute faster (than an interpreted source code program) and that the compiler will not need to be present when the object code is executed.

Part **(c)** was well answered with a large number of possible points which could score the marks. Wrong answers included a description of syntax checking using the BNF rules and other processes which are not carried out until the syntax analysis stage.

Question 5

This question was commented on earlier in the General section. The most common error was candidates who did not focus on the word 'application' in the rubric of part **(c)** and simply described the push and pop operations of any stack data structure.

Question 6

Candidates needed to develop their understanding for this question as answers seen for part **(a)** were generally weak with candidates unable to say more than 'the memory will be free-up'. Better answers conveyed the idea that a 'hole' would be created of available memory and either it will be 'self-contained' or could, if adjacent to an existing hole, create a larger hole. Alternatively when a program terminates all programs could be re-located to create one large hole of available space.

Answers for part **(a)(ii)** were dependant on what the candidate had suggested for part **(i)**. Answers required would make maximum use of the available memory space and then use the high-level scheduler to decide which process(es) to load next.

Part **(b)** proved to be a good discriminator. Candidates generally scored with their description of an interrupt and then had to name and describe sources of an interrupt. Most common answers were to name a device (e.g. a printer) and then give the reason (e.g. that its ink levels were low). Other devices used were a CD-drive, memory stick and others. A named source of 'hardware' or 'I/O device' was considered insufficient. Other worthy answers named the source as 'software' (which was considered sufficient) and then a valid reason (such as a division by zero error or failure to find a device, etc.).

For part **(b)(iii)** the answer given depended to some extent on the nature of the interrupt. The fundamental steps common to all would have included: identify the source of the interrupt, prioritise or disable lower priority interrupts, save the register contents, run the ISR, retrieve register contents, enable all interrupts, continue with the interrupted process.

Question 7

Candidates were sometimes unclear of what was meant by the term 'media'. Many candidates confused the media with their description which followed, but were still given credit. Some candidates did not take heed of the word 'different' in the rubric and described two different types of cabling.

Candidates were required to be resourceful for part **(b)** and most scored well. The key items of hardware expected were the inclusion of the two file servers and some form of communication from Shop B and Shop C using a router and/or modem.

Most candidates could give benefits of the company having an intranet, but few realised that the intranet pages and content would be stored and made available from a web server.

Question 8

Both parts were generally well answered with correct answers for part **(a)(i)** and part **(ii)** and candidates realising that the expression in part **(iii)** was not properly formed.

Question 9

There were many points which could score the two available marks for part **(a)**. The most popular answers stated that the program is made up of a sequence of instructions/statements, that all the languages use the three basic constructs of assignment, selection and iteration. Weaker answers – which still scored – described the program solution following a top-down design and hence being coded as a series of procedures.

Part **(b)** was well answered with many candidates scoring the full four marks.

COMPUTING

Paper 9691/32
Written Paper

General

The performance of the candidates on this paper was very varied. It is appreciated that for the majority of candidates English is not their first language and wherever possible the Examiners will take account of this. However for some questions a clear statement is required to gain the available marks. An example of this was **Question 3(a)** parts **(ii)** and **(iii)** where candidates were rarely able to convey their understanding. The quality of answers for **Question 5(b)** has been highlighted in a previous examination report as a weakness. Answers were often seen where data values were shown at one or more of the missing steps in the algorithm. It was indicative of candidates' preparation for the examination where they were usually able to describe the operation of a stack data structure - **Question 5(a)** – but then were rarely able to give a practical application of the use of a stack data structure in a computer system.

Question 1

This question proved to be a good discriminator. A number of candidates gave a clear and accurate answer to part **(a)(i)** but then needed to expand on their answer for part **1(a)(ii)**. Candidates were required to be expansive for part **(b)(i)** and found no problem in coming up with appropriate attributes for the course and employee table. The usual mistake was to include attributes in the employee table which related to course attendance.

Candidates needed to improve their understanding for this question as part **(ii)** rarely gained full marks. Candidates often appreciated that the CourseCode - or their primary key from part **(i)** - and the EmployeeID would be present, but then did not realise these two attributes would form the composite primary key for table CourseEnrolment.

Question 2

Candidates who had researched past examination papers would have seen a familiar question framework; however, a significant number of candidates needed to improve on their responses to parts **(b)(i)** and **(ii)**. Often candidates correctly computed the exponent to be 7, but were then unable to compute the mantissa for part **(iii)**. Candidates who moved the binary point of the mantissa seven places to compute the final answer -120 gained full marks.

Question 3

Part **(a)(i)** was intended as a straightforward mark with candidates demonstrating their knowledge of the registers MDR and CIR. Candidates often lost the mark by not expanding on one or both of the acronyms. Please refer to comments in the general section relating to parts **(a)(ii)** and **(iii)**. Part **(b)** was much better answered with candidates able to score. Part **(c)** was often well answered with a description of direct addressing. Candidates who followed the suggestion of a diagram often scored the full 2 marks, showing a block of memory with an arrow pointing to a particular address and then its contents being copied to a register. Relative addressing was not well understood. This is an important concept which is the basis of code being re-locatable. Candidates who tried to guess usually gave a wrong answer describing indexed, indirect or base register addressing (which is not in the syllabus).

Question 4

This question was a good differentiator and the majority of candidates were able to score. A not uncommon incorrect answer seen for part **(a)** was the suggestion that object code is different to machine code, so these given as outputs scored only 1 of the available 2 marks. A common error for part **(b)** was a wording in the answer which did not make it clear whether the answer 'faster' referred to the compilation translation process or execution of the final object code.

Question 5

Please refer to the earlier comments in the 'General' section. Candidates' ability to complete the steps for a given algorithm needs further development. This is a skill which should be practiced with candidates, often as a pre-cursor to writing program code. ***Section 3.4*** of the syllabus describes the range of data structures for which work of this nature could be done.

For part **(c)(i)** candidates often simply described the operation of a stack and did not focus on the word 'application'. Answers expected were the use of a stack for managing procedure calls, interrupt handling or the evaluation of a reverse Polish expression.

Question 6

A large number of candidates scored the full four marks for naming and describing paging and segmentation. Other worthy answers described partitioning and the use of virtual memory. Weaker candidates sometimes confused memory management with processor management and answers such as 'round robin' were incorrectly given as answers for part **(a)**. Part **(c)** showed a clear differentiation of answers. Candidates sometimes confused the 'source' and the 'reason' but were still given credit. An imprecise answer such as the source as 'hardware' or an 'I/O interrupt' was considered insufficient.

Question 7

Candidates were sometimes unclear about what was meant by the term 'media'. Many candidates confused the media with their description which followed, but were still given credit. Some candidates did not take heed of the word 'different' in the rubric and described two different types of cabling.

Part **(b)(ii)** was intended as a straightforward three marks with; a star shaped topology with three branches with a computer at the end of each, labelling "Town A", etc. and the labelling of the computer at the centre of the star.

For part **(b)(iii)** credit was not given for points which were general benefits of any network topology. The most popular answer which scored was that *"if one terminal/communication line fails other branches are not affected"*.

Question 8

This question showed good differentiation but with answers for parts **(i)** to **(iv)** often being weak. This should clearly be taught in a practical way with candidates exposed to content and skills they will also need for Paper 2. Few correct answers were seen for part **(b)(iii)**. Candidates were expected to identify the first parameter as a 'string' data type (and not the required 'character' data type).

Question 9

Many candidates lost marks due to insufficient care taken in the use of case for the data and clause names (both of which must be lower case).

Parts **(a)** and **(b)** generally scored well but candidates needed to improve their answers for part **(c)**. Candidates often showed some understanding, e.g. the use of the country clause twice with a common variable, but were unable to successfully put the ideas together into a correctly formed rule.

COMPUTING

Paper 9691/33
Written Paper

General

The performance by candidates showed a large range of abilities. Candidates are often able to reproduce basic bookwork type definitions but are then less able to either: apply this knowledge to some given scenario – e.g. **Question 5(c)**, or use the knowledge for some skills task – e.g. 5**(b)(i)**. Candidates appear to remain weak at the understanding of a given algorithm and responses are still being seen which have data values added as part of an incomplete algorithm. Candidates need to appreciate that there is overlap here with the style of question in **Question 5(b)** and their practical programming work undertaken for Paper 2.

Question 1

Part **(a)(i)** was well answered but for part **(ii)** candidates needed to develop better understanding as they were often unable to explain how a primary key is used with a foreign key to form a relationship. A concise statement such as *"The primary key in table X links to a foreign key attributes in Table Y"* would have secured the full 3 marks.

For part **(b)(i)** candidates had to be resourceful and come up with additional attributes for the Book and Student table and found no problem in doing this. For part **(ii)** candidates correctly included the BookID and StudentID as attributes but needed further understanding to work out from the rubric the correct primary key.

Some clear answers were seen for part **(c)** which often explained the term using an example.

Question 2

Parts **(a)** and **(b)** were generally well answered. The explanation expected for part (**b)(i)** was either that overflow has occurred or that the answer computed is outside the range of numbers possible using 8-bit 2's complement.

Question 3

Part **(a)(i)** was intended as a straightforward mark, but candidates needed to improve on their answer by expanding the acronyms into Current Instruction Register and Memory Data Register. Answers seen for part **(ii)** were weak. It was suspected that this was the result, not of a lack of understanding, but incomplete expression by the candidate. An answer for part **(ii)** was probably expressed most clearly with two sentences: *"The Memory Address Register holds the address to be used. The data value at this address is copied to the Memory Data Register".*

Part **(b)** was well answered with candidates able to score for stating the basic point that the *"the assembler translates assembly language code into machine/object code"*. This was then followed with points of detail such as the looking up of all the opcode mnemonics in an opcode table and building a symbol table of addresses. The detail of one-pass and two-pass assemblers was not expected.

Candidates need to further develop their understanding of relative addressing. They need to be familiar with this mode of addressing as it forms the basis of the ability to re-locate code when this is covered in the syllabus section on memory management.

For part **(c)(i)** better answers that were seen followed the suggestion of a diagram which typically showed a block of memory with an arrow pointing to the address to be used and its contents then circled and copied to a register.

Question 4

Part **(a)** was well answered, the common answers being the executable/object code and some form of error report. Some answers gave object code and executable code implying (incorrectly) that they were somehow different.

The interpretation process was well understood, although some candidates wrongly stated that object code is produced by the interpreter.

Most candidates scored for part **(c)**. The most common mistake was to suggest tasks which are done at the lexical analysis stage.

Question 5

Please refer to earlier comments about this question. The three missing steps of the AddToQueue algorithm were to test that the queue was not already full (TailOfQueue = 101) and to assign the new data item to the correct position in the MyQueue array (MyQueue(Tail) ← NewItem). This must then be followed by incrementing TailOfQueue.

Writing the RemoveFromQueue algorithm was more demanding and candidates struggled to do this without a similar framework to part **(i)**.

For part **(c)** the most popular answer was the creating of a spool queue for print jobs. Candidates sometimes were unable to score the second mark for any justification that their given application would need to function as a queue data structure.

Question 6

Part **(a)** was well answered. The two popular answers were to allocate a fixed size time slice to each process and then process each in turn ('round-robin') or to allocate some form of priority to each process. For the idea of priority the candidates had to state what the priority was based upon (e.g. a user priority when the task is first scheduled, or based on the estimated shortest remaining run-time). It was insufficient to make an unclear statement such as 'shortest job next" (which was a common answer).

Part **(b)** the most common answers were to name a device (e.g. a printer) and then give the reason (e.g. that its ink levels were low). Other devices used were a CD-drive, memory stick and others. A named source of 'hardware' or 'I/O device' was considered insufficient. Other worthy answers named the source as 'software' (which was considered sufficient) and then a valid reason, such as a division by zero error or failure to find a device, etc.

For part **(b)(ii)** the answer given depended to some extent on the nature of the interrupt. The fundamental steps common to all would have included: identify the source of the interrupt, prioritise or disable lower priority interrupts, save the register contents, run the ISR, retrieve the register contents, enable all interrupts, continue with the interrupted process.

Question 7

Candidates were sometimes unclear about what was meant by the term 'media'. Many candidates confused the media with their description which followed, but were still given credit. Some candidates did not take heed of the word 'different' in the rubric and described two different types of cabling.

Again candidates had to be resourceful in drawing a diagram which should have included: a single cable run, terminators added, four computers attached and with a printer attached to one of the computers.

For part **(b)(ii)** candidates could usually mention the two key points that a WAN consists of a number of computer/networks which are connected over a large geographical area (e.g. country-wide).

Candidates who had seen previous examination papers would have found no problem with this question in naming and describing three security measures.

Question 8

This question was well answered with many candidates scoring the four available marks for evaluating the functions in parts **(ii)** to **(v)**.

For part **(b)** the most common correct answer was to state that a function always returns a value – a procedure may/may not return value(s).

Question 9

This question was answered well in parts **(a)** and **(b)**. Most inheritance diagrams correctly showed a parent class ACCOMODATION with sub-classes of SINGLEROOM and SUITE pointing to the parent.

Candidates needed to develop their understanding for part **(c)**. The concept looked for was that property values are only accessed through methods available in the class definition for the reading and writing of data. An example could have been given; for example, a RoomNo property would require both a 'getRoomNo' and 'setRoomNo' method in the ACCOMODATION class.

COMPUTING

Paper 9691/04

Project 2

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.

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. 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 for example feasibility studies and cost benefit analysis.

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 re-typed/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 37 of the 2012 Syllabus.

Project assessment and marking

All 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 described the methods used and also described the origin of the data and indicated the 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 study of documents currently in use; then ask follow up questions to fill in any gaps in the knowledge obtained about the current system or requirements needed for the new system. 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.

In order to obtain good marks for this sub-section, candidates need to 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

In order to obtain good marks for this sub-section, 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)** and be annotated by the candidate.

(ii) Programming

It is important that the programming code in this subsection is written by the candidate and not produced as a result of tailoring a software package. Marks should only be awarded to code that has been written by the candidate.

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 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. Many candidates only tested the validation and navigation aspects of their system, and omitted to test that their system did what it is supposed to do, for example production of reports. This omission meant candidates were unable to gain marks in the highest band for this sub-section.

(iv) Installation

Most candidates provided a detailed implementation plan containing details of user testing, user training and system changeover.

For good marks to be awarded written evidence from the client and/or user(s) must be included in order to show that the system has been seen and used and the candidate's plans have been agreed.

Centres are reminded that appropriateness of structure and exploitation of available facilities are not now required for this sub-section of the report.

(e) Documentation

(i) Systems Maintenance Documentation

This sub-section of the report is now a systems maintenance document. Please see page 43 of the 2012 Syllabus for details of what should be included.

For top marks to be awarded the candidate must explain how adaptive maintenance could be undertaken.

(ii) User Documentation

This section was completed to a good standard by most candidates. Centres are 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.

(f) Evaluation

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

(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

A response must be provided directly 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.

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