COMPUTER SCIENCE

Paper 9608/11
Written Paper

Key messages

The title of this qualification is now Computer Science and to succeed in this paper it is essential that candidates are able to make correct use of appropriate technical terminology. The subject includes a large number of technical terms or computer jargon and it will be expected that these technical words will be used in the correct context when answering questions.

Databases and Data Modelling have appeared on this paper for the first time this session and the best way to prepare candidates for questions on this topic is by exposing them to some practical work with a database prepared and made available by the teacher together with a software tool which allows the interrogation of the data by the candidate.

There is a great deal of very good information online about current developments in Computer Science, and candidates are clearly making excellent use of these resources. However, candidates should be encouraged to ensure that the articles used for learning and revision are both from a reliable source and are at the appropriate academic level.

General comments

Questions on this paper, such as **Question 2(a)(iv)**, **Question 2(c)** and **Question 8(b)** often require explicit or implicit comparisons; at this level of study these comparisons must be justified. It is insufficient to say, for example, "device A is better/faster/cheaper than device B". There needs to be some justification of why device A is better or faster or cheaper than device B.

Comments on specific questions

Question 1

It is appreciated that candidates will need space to work out the answer to questions such as **Question 1** and so there is blank space below the question on the examination paper. Some candidates put their working in the answer space and obscured the answer. The number which is to be marked should be clearly visible.

(i) The majority of candidates correctly converted the binary value to hexadecimal and many also correctly included the subscript and gave the answer as $B8_{16}$. The most common error was to convert the binary integer to 184 in denary, but to then not convert that denary value to hexadecimal.

(ii) Many candidates recognised what was required in Binary Coded Decimal (BCD) format and correctly converted 97 as a denary integer into 1001 0111 in BCD. The most common mistake here was to not recognise that BCD format was required and complete a straightforward conversion from 97 in denary to 0110 0001 in two's complement binary.

(iii) In general, most of the answers to this part question were correct, although the more difficult skill of converting the negative number was sometimes not correctly completed.

Question 2

Please see the comments in the **General** section of this report regarding comparisons.

(a) (i) Whilst most candidates seemed to appreciate that this term was used in the conversion of some analogue data to digital there is a need for improvement in their understanding of exactly what sampling means. Responses which simply re-worded the question stem were common, such as '*sampling is the number of samples taken'*; this is much too imprecise when the question asks for a definition.

(ii) This question proved challenging for many candidates. There was a general misconception that a CD could not hold anything sampled at a resolution greater than 16 bits rather than 16-bit sampling being a compromise between the accuracy of the sampled sound and hence the quality of the playback and the amount of disk space required to store the sound file and hence the number of files which could be stored on a CD of fixed size.

(iii) There was considerable confusion between sampling resolution, the number of distinct values available to encode each sample, and sampling rate, the number of samples taken per unit time. The question also asked for an explanation so needed some expansion of the initial statement.

(iv) The majority of candidates realised that improving sampling resolution resulted in a more accurate representation of the sound, and that as a consequence of this the file size would be increased. The written responses though were often too imprecise. See the comments in the **General** section above.

(b) The features of sound editing software identified were many and varied and usually correct. Care must be taken to ensure that features such as cropping are given in the context of the sound editing software and not generically.

(c) This question was very well answered with many candidates giving detailed answers including descriptions of compression methods and correct examples of compressed file types.

Question 3

This was well answered. A few candidates confused immediate addressing and indirect addressing.

Question 4

(a)(i)(ii)(iii) There were many correct answers here, with the sensors usually correctly identified. A small number of candidates put measuring devices rather than sensors.

(b) Most candidates seemed to understand methods of ensuring the integrity of the data during transmission. The question asked for a description, these varied in quality. Some were very good and included all the necessary detail others were less detailed and sometimes omitted important details such as when using a checksum the same calculation needs to be completed at the receiving end. Descriptions of a parity check were sometimes not clear, with candidates confusing parity bits and parity bytes.

Question 5

(a) Some candidates completed the table correctly; most candidates correctly completed the last three rows. A common mistake was for candidates to not follow the instruction in the question which stated '*Put a tick...*' and put two ticks in the same row.

(b) Candidates need to improve their understanding of the difference between the World Wide Web (WWW) and the Internet. Most candidates understood that the Internet was a network, but there was little appreciation of the scale of the connection. Quite a few candidates confused the WWW with a sub-section of the Internet.

(c)(i)(ii)(iii) Candidates need to improve their knowledge of the hardware devices used in networks. Common errors were to confuse routers with hubs and modems, and to confuse gateway with router.

Question 6

(a) The majority of candidates answered this question well. A common error was the omission of the type of printer on the third and fourth rows. At this level, candidates are expected to be able to differentiate between different types of printers and identify which type would be suitable for any given task.

(b) (i) The question states '*describe*' and some candidates were able to describe one method of preventing accidental loss of data, but few provided two correct descriptions. Many candidates overlooked the word 'accidental' in the question and so gave incorrect answers. It is not enough to provide single word answers for a description. Candidates need to improve their understanding of the backup process, which needs to be completed regularly, and backups stored safely.

(ii) This question also states '*describe*' and many candidates were able to describe a method of ensuring data security. Single word answers are not enough for a description. Candidates need to improve their understanding of security methods, it is not enough to install a firewall, for example, it needs to be switched on and operational, and similarly simply having anti-virus software is not enough, the virus definitions need to be regularly updated if the software is to be effective.

Question 7

(a) This question was very well answered and there has been considerable improvement in the neatness and clarity of the logic diagrams. The most common error was the omission of the NOT gates on some inputs.

(b) This question was very well answered

Question 8

(a) Candidates need to improve their understanding of the components of the processor. Only a minority of candidates were able to explain how the width of the data bus and the clock speed affected processor performance. Many answers were too vague and imprecise. Some candidates confused width of the bus with length. Explanations of clock speed often omitted any reference to time.

(b) Candidates need to read the question carefully. The question asked for the benefits of using USB ports, the majority of candidates gave answers about USB devices.

(c) This question was very well answered with most candidates able to give a correct sequence of stages in the fetch-execute cycle.

Question 9

(a) A few candidates answered this question correctly, but there is a need for greater precision in the use of the terminology.

(b) This question was very well answered with almost all candidates able to complete the tables correctly.

(c) (i) The majority of candidates answered this question correctly. The most common error was to not identify the common attribute.

(ii) Many candidates were able to explain that the table was not in Third Normal Form because a non-key dependency existed, common errors were to fail to identify the non-key dependency or to incorrectly identify it. Other candidates also correctly stated that the table was not in Second Normal Form, but omitted to say why this was the case.

(iii) A few candidates correctly wrote the table definitions. In general, candidates need to improve their understanding of written table definitions. A common error was to omit the identification of the primary keys in the tables.

COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

The title of this qualification is now Computer Science and to succeed in this paper it is essential that candidates are able to make correct use of appropriate technical terminology. The subject includes a large number of technical terms or computer jargon and it will be expected that these technical words will be used in the correct context when answering questions.

Databases and Data Modelling have appeared on this paper for the first time this session and the best way to prepare candidates for questions on this topic is by exposing them to some practical work with a database prepared and made available by the teacher together with a software tool which allows the interrogation of the data by the candidate.

There is a great deal of very good information online about current developments in Computer Science, and candidates are clearly making excellent use of these resources. However, candidates should be encouraged to ensure that the articles used for learning and revision are both from a reliable source and are at the appropriate academic level.

General comments

Questions on this paper, such as **Question 2(a)(iv)**, **Question 2(c)** and **Question 8(b)** often require explicit or implicit comparisons; at this level of study these comparisons must be justified. It is insufficient to say, for example, "device A is better/faster/cheaper than device B*"*. There needs to be some justification of why device A is better or faster or cheaper than device B.

Comments on specific questions

Question 1

It is appreciated that candidates will need space to work out the answer to questions such as **Question 1** and so there is blank space below the question on the examination paper. Some candidates put their working in the answer space and obscured the answer. The number which is to be marked should be clearly visible.

(i) The majority of candidates correctly converted the binary value to hexadecimal and many also correctly included the subscript and gave the answer as $B8_{16}$. The most common error was to convert the binary integer to 184 in denary, but to then not convert that denary value to hexadecimal.

(ii) Many candidates recognised what was required in Binary Coded Decimal (BCD) format and correctly converted 97 as a denary integer into 1001 0111 in BCD. The most common mistake here was to not recognise that BCD format was required and complete a straightforward conversion from 97 in denary to 0110 0001 in two's complement binary.

(iii) In general, most of the answers to this part question were correct, although the more difficult skill of converting the negative number was sometimes not correctly completed.

Question 2

Please see the comments in the **General** section of this report regarding comparisons.

(a) (i) Whilst most candidates seemed to appreciate that this term was used in the conversion of some analogue data to digital there is a need for improvement in their understanding of exactly what sampling means. Responses which simply re-worded the question stem were common, such as '*sampling is the number of samples taken'*; this is much too imprecise when the question asks for a definition.

(ii) This question proved challenging for many candidates. There was a general misconception that a CD could not hold anything sampled at a resolution greater than 16 bits rather than 16-bit sampling being a compromise between the accuracy of the sampled sound and hence the quality of the playback and the amount of disk space required to store the sound file and hence the number of files which could be stored on a CD of fixed size.

(iii) There was considerable confusion between sampling resolution, the number of distinct values available to encode each sample, and sampling rate, the number of samples taken per unit time. The question also asked for an explanation so needed some expansion of the initial statement.

(iv) The majority of candidates realised that improving sampling resolution resulted in a more accurate representation of the sound, and that as a consequence of this the file size would be increased. The written responses though were often too imprecise. See the comments in the **General** section above.

(b) The features of sound editing software identified were many and varied and usually correct. Care must be taken to ensure that features such as cropping are given in the context of the sound editing software and not generically.

(c) This question was very well answered with many candidates giving detailed answers including descriptions of compression methods and correct examples of compressed file types.

Question 3

This was well answered. A few candidates confused immediate addressing and indirect addressing.

Question 4

(a)(i)(ii)(iii) There were many correct answers here, with the sensors usually correctly identified. A small number of candidates put measuring devices rather than sensors.

(b) Most candidates seemed to understand methods of ensuring the integrity of the data during transmission. The question asked for a description, these varied in quality. Some were very good and included all the necessary detail others were less detailed and sometimes omitted important details such as when using a checksum the same calculation needs to be completed at the receiving end. Descriptions of a parity check were sometimes not clear, with candidates confusing parity bits and parity bytes.

Question 5

(a) Some candidates completed the table correctly; most candidates correctly completed the last three rows. A common mistake was for candidates to not follow the instruction in the question which stated '*Put a tick...*' and put two ticks in the same row.

(b) Candidates need to improve their understanding of the difference between the World Wide Web (WWW) and the Internet. Most candidates understood that the Internet was a network, but there was little appreciation of the scale of the connection. Quite a few candidates confused the WWW with a sub-section of the Internet.

(c)(i)(ii)(iii) Candidates need to improve their knowledge of the hardware devices used in networks. Common errors were to confuse routers with hubs and modems, and to confuse gateway with router.

Question 6

(a) The majority of candidates answered this question well. A common error was the omission of the type of printer on the third and fourth rows. At this level, candidates are expected to be able to differentiate between different types of printers and identify which type would be suitable for any given task.

(b) (i) The question states '*describe*' and some candidates were able to describe one method of preventing accidental loss of data, but few provided two correct descriptions. Many candidates overlooked the word 'accidental' in the question and so gave incorrect answers. It is not enough to provide single word answers for a description. Candidates need to improve their understanding of the backup process, which needs to be completed regularly, and backups stored safely.

(ii) This question also states '*describe*' and many candidates were able to describe a method of ensuring data security. Single word answers are not enough for a description. Candidates need to improve their understanding of security methods, it is not enough to install a firewall, for example, it needs to be switched on and operational, and similarly simply having anti-virus software is not enough, the virus definitions need to be regularly updated if the software is to be effective.

Question 7

(a) This question was very well answered and there has been considerable improvement in the neatness and clarity of the logic diagrams. The most common error was the omission of the NOT gates on some inputs.

(b) This question was very well answered

Question 8

(a) Candidates need to improve their understanding of the components of the processor. Only a minority of candidates were able to explain how the width of the data bus and the clock speed affected processor performance. Many answers were too vague and imprecise. Some candidates confused width of the bus with length. Explanations of clock speed often omitted any reference to time.

(b) Candidates need to read the question carefully. The question asked for the benefits of using USB ports, the majority of candidates gave answers about USB devices.

(c) This question was very well answered with most candidates able to give a correct sequence of stages in the fetch-execute cycle.

Question 9

(a) A few candidates answered this question correctly, but there is a need for greater precision in the use of the terminology.

(b) This question was very well answered with almost all candidates able to complete the tables correctly.

(c) (i) The majority of candidates answered this question correctly. The most common error was to not identify the common attribute.

(ii) Many candidates were able to explain that the table was not in Third Normal Form because a non-key dependency existed, common errors were to fail to identify the non-key dependency or to incorrectly identify it. Other candidates also correctly stated that the table was not in Second Normal Form, but omitted to say why this was the case.

(iii) A few candidates correctly wrote the table definitions. In general, candidates need to improve their understanding of written table definitions. A common error was to omit the identification of the primary keys in the tables.

COMPUTER SCIENCE

Paper 9608/13
Written Paper

Key Messages

The title of this qualification is now Computer Science and to succeed in this paper it is essential that candidates are able to make correct use of appropriate technical terminology. The subject includes a large number of technical terms or computer jargon and it will be expected that these technical words will be used in the correct context when answering questions.

There is a great deal of very good information online about current developments in Computer Science, and candidates are clearly making excellent use of these resources. However, candidates should be encouraged to ensure that the articles used for learning and revision are both from a reliable source and are at the appropriate academic level.

General Comments

Questions on this paper, such as **Question 3(b)**, **Question 4(b)** and **Question 4(c)** often require explicit or implicit comparisons; at this level of study these comparisons must be justified. It is insufficient to say, for example, "device A is better/faster/cheaper than device B". There needs to be some justification of why device A is better/faster/cheaper than device B.

In some questions, for example **Question 3(b)**, it is essential that candidates use the correct words in descriptions as alternatives can have a slightly different meaning which makes the description incorrect.

Comments on Specific Questions

Question 1

It is appreciated that candidates will need space to work out the answer to questions such as **Question 1** and so there is blank space below the question on the examination paper. Some candidates put their working in the answer space and so obscured the answer. The number which is to be marked should be clearly visible.

(a) (i) In general, most of the answers to this part question were correct, although the more difficult skill of converting the negative number was sometimes not correctly completed.

(ii) The majority of candidates correctly converted the binary values from **(a)(i)** to hexadecimal and many also correctly included the subscript and gave the answers as $7C_{16}$ and $B3_{16}$. Some candidates did not convert from binary but from the original denary values which was an acceptable alternative.

(b) (i) Many candidates recognised what was required in Binary Coded Decimal (BCD) format and correctly converted 359 as a denary integer into 0011 0101 1001 in BCD. The most common mistake here was to not recognise that BCD format was required and complete a straightforward conversion from denary to two's complement binary.

(ii) Candidates need to improve their understanding of the use of BCD number representation. The most common error was that BCD numbers are used in bar codes.

Question 2

Candidates need to improve their understanding of two-pass assemblers. Most candidates realised that the activity on the fourth line would need to be completed during the first pass, but there was considerable confusion about the other stages. Some candidates wrote that all the activities were carried out in the same stage.

Question 3

Please see the comments in the **general comments** section of this report regarding comparisons.

(a) Most candidates could define the two terms, but some found difficulty in explaining how either a firewall or authentication helped with data security. Explanations were sometimes vague and imprecise.

(b) The majority of candidates were able to describe data security, but many found it harder to describe data integrity. There seems to be a misconception that data integrity refers to the accuracy or correctness of data rather than the validity of the data. Valid data may not be accurate.

(c) (i) There were many excellent examples given in response to this question, some candidates did not explicitly name validation and verification.

(ii) As in **(c)(i)** there were many excellent examples, although some candidates did not name the method. The method should be named when the question states '*State*'.

Question 4

(a) This question was quite well answered. A few candidates confused the two types of RAM and so their answers were completely back to front. The most common error was that DRAM had the more complex circuitry.

(b) This question was very well answered.

(c) Most candidates knew that DVD-RAM was a form of optical storage using laser technology; however, candidates need to improve their understanding of the operation of flash memory. Responses were generally too vague and not precise or detailed enough for A Level. Also please see the statement about comparisons in the **General** section of this report.

Question 5

(a) The majority of candidates could name the three buses used in the von Neumann model; describing them proved more challenging. Descriptions were often too vague and did not include sufficient detail about either the direction of the bus, or what the bus was used for. Often the description of the control bus omitted to include anything about control signals.

(b)(i)(ii)(iii) In questions such as this it is essential that candidates use the correct terminology. Many candidates did this and these questions were mostly well answered.

(c) This question was very well answered.

(d) (i) Many candidates answered this question very well, although there needs to be better understanding that an interrupt is a signal, not a message, and why interrupts are generated.

(ii) Candidates need to improve their understanding of how an interrupt affects the actions of the processor during the fetch-execute cycle. Some candidates did know that the contents of the registers needed to be saved although many only wrote about saving the contents of the Program Counter (PC) and ignored the other registers; few candidates mentioned an interrupt flag or interrupt register. Only a small number of candidates included loading the address of the Interrupt Service Routine into the PC.

Question 6

(a) This question was very well answered.

(b) This question was very well answered.

(c) This question was very well answered. A small number of candidates did not write a logic statement.

Question 7

(a) (i) This question was very well answered. The majority of candidates knew how to apply direct addressing.

(ii) This question was very well answered; the majority of candidates also knew how to apply indirect addressing.

(iii) This question was well answered but a number of candidates incorrectly added the contents of the index register (6_{10}) to the contents of address 120 (9_{10}) to give a value of 15_{10} which appeared in the Accumulator as 0000 1111.

(b) This question was very well answered. The majority of candidates were able to correctly trace the assembly language program and enter the correct values in the appropriate places in the trace table.

COMPUTER SCIENCE

Paper 9608/21
Written Paper

General comments

This was the first examination sitting for this new 9608 Computer Science syllabus.

For this Paper 2 candidates were expected to present for the examination having previously studied the Pre-release materials which had been circulated to centres. There were some excellent answers for the questions where the candidate was required to write program code in their chosen language, but there were a significant number of scripts where no attempt was made by the candidate for questions which required the writing of code and clearly such candidates were at a significant disadvantage. Candidates need extensive practical programming experience prior to sitting for the examination. There was little programming skill exhibited by a significant number of candidates.

The questions paper asked the candidate on page 2 to write the programming language they would be using (Visual Basic.NET, Python or Pascal). In the questions which required the writing of program code, the candidates was instructed to write the programming language used on the first dotted response line. Many candidates did not do this. The majority of candidates used Visual Basic, followed by Python, with a small minority using Pascal. Candidates need to be clear when reading the question if it is pseudocode or program code they are being asked to write. Some candidates stated 'pseudocode' as their programming language which did not gain credit.

The syllabus gives very detailed guidance on the key words to use when writing or completing a pseudocode algorithm. Candidates need to appreciate when it is appropriate to use the assignment operator or the '**=** ' symbol to test for equality.

Question 1

Part **(a)** was well answered with the majority of candidates gaining the full three marks. Candidates need to make sure that when they are asked to give an identifier name that it does read as such, with no <Space> characters. All too often the identifiers read like text – for example 'race time seconds' and not RaceTimeSeconds, or even worse 'RaceTime (Seconds) which suggests an array. This was an issue which carried through into later questions. Candidates need to follow some simple rules i.e. no space or inappropriate characters and adopt a consistent style such as CamelCase.

Part **(b)(i)** was well answered with a sensible name used for the personal best time. Data type INTEGER was expected. Some candidates had suggested in part **(a)**, the seconds input time be a real number (for example, time measured to the nearest hundredth of a second) and so declared the personal best time as REAL which gained credit.

For part **(b)(ii)** candidates gained the first two mark points for declaring variables and inputting the hours, minutes and seconds time. Most answers also followed this with the input of the personal best time. It was encouraging to see that candidates who wrote a Python solution did follow the question and include a comment statements giving the data type used for a variable.
Calculation of RaceTimeInSeconds was usually correct. Some candidates did not use the correct symbol for 'multiply' and did not gain credit.
Many answers omitted from their code the output of the RaceTimeInSeconds.
The logic for the last three mark points was widely misunderstood. It was a widespread error that if the race time was larger than the personal best time then this would general a new personal best time. Candidates need to be clear about the meaning of the <, >, and <> operators and their application.

For part **(c)(i)** the meaning of white-box testing was not well understood. The two key points looked for were "that the programmer would need to choose data values carefully" and "in order to test every path through the code and produce every possible outcome". Candidates needed to make it clear that the white-box tests carried out would assume knowledge of the particular code for this program. Some candidate answers read as if what was needed was the general skill of program coding.

Part **(c)(ii)** seemed to attract either very good or very weak responses.

The misunderstanding about whether or not the race time would or would not produce a new personal best carried through into this question about the selection of test data. Key points looked for were the selection of a dataset (hour, minutes and seconds) which was faster that the current personal best, followed by a dataset which gave a slower time, or vice versa. The final two marks were then for the display of the appropriate message. Candidates do need to be reasonably realistic with their data choice.
A significant number of candidates stated twice the same time values as given in the question on row one and the intention was unclear.

Question 2

For part **2(a)(i)**, most candidates received the mark for a statement about the input needing to be 'between 1 and 4'. The key wording in the question was 'what this pseudocode would do' and so a reference to "calling DisplayMenu" did not gain credit, whereas "displaying the menu choices to the user" scored.
Few candidates secured the second mark as they usually did not refer to the prompt.
Being able to clearly express an algorithm is a key skill for the computer scientist. This skill was again required later to secure the marks for **Question 5(b)(ii)**.

For part **(a)(ii)**, candidates should appreciate that the purpose of the algorithm was to 'validate' the input from the user.

Part **(b)(i)** was well answered. Most candidates correctly stated there would be three iterations of the loop.

For **part (b)(ii)**, most candidates secured the mark for stating that there would be a limit on the number of attempts allowed.

Part **(c)** was generally well answered. Common errors included the use of a different variable name – not Choice - and outputting the string "Call ReadFile" instead of pseudocode for a call to the ReadFile procedure. Answers using a Case structure were less likely to be correct than those using IFs. This was one of the questions where candidates did not gain any marks if they had incorrectly used the assignment operator ← instead of =.

Answers for the program code solution for part **(d)** were varied. It was good to see that candidates using Python did take note of the question and included a comment statement to state the data type used for their variables. However, some candidates read this as the cue to comment just about every line of their code, which was not the intention.
For Visual Basic and Pascal, the correct syntax was required for the declaration and assignment for constant i. Python candidates were only expected to show a normal assignment statement.
The mostly commonly gained marks were for:

- the declaration of one of the Integer variables
- the input of the choice value
- a correctly formed selection for the Choice value.

Code for the ReadFile and DisplayMenu procedures was often omitted and there were few solutions seen which had a correctly formed loop which included a call to DisplayMenu and which terminated with a user entry of 4.

Question 3

The majority of candidates scored well on part **(b)** but not for part **(a)**. Some candidates did not follow the question and wrote the module and data item descriptions on the dotted response lines (not the letter labels). A common error in part **(a)** was to label B as the control box.

Question 4

The majority of candidates achieved at least 3 marks for this question.

A common error was using the keyword NEXT instead of ENDFOR. Some answers confused the concept of a player 'throw' and a player 'total'. This misunderstanding was sometimes carried forward into the explanation for part **(ii)**. For part **(ii)**, candidates realised that if the both player totals were 10 then an inappropriate message was output.

Question 5

The syllabus distinguishes between a 1D and 2D array and so there was an expectation that the answers for part **(a)** should describe a 'ID array'. The second mark was awarded for a stated data type of integer.
The word 'array' was often mentioned, but frequently without the '1-D' prefix. 'Integer' was seen frequently.
There were many incorrect answers which included a specific pseudocode array declaration statement or mention of different structures (stack and queue).

The algorithm trace for part **(b)(i)** was generally well answered with a significant number of maximum marks.
Common errors included the omission of the final output of '3' and additional values present in the OUTPUT column. Less able candidates sometimes secured the mark for the values 1-7 in the DayNumber column, but then the first column values of 1, 2 and 3 (although correct) were placed on the incorrect row of the table and so did not gain credit. Candidates should be made aware that this horizontal alignment is significant when completing a trace table.

For part **(b)(ii)**, few candidates achieved the first mark point for the phrase 'first seven days'.
Most answers gained the second mark for some reference to total sales being 10 or more.
Output of the 'day' was often incorrectly stated in place of 'date' and output of the final total was often omitted. See the earlier general point made for **Question 2(a)(i)**.
For the table in part **(b)(ii)** candidates generally secured the mark for the first and third row. The second function call was often correctly identified as incorrect, but the explanation was insufficient and lacked any reference to 'parameters, for example, *"*only one character can be entered" or *"*can only look at sales from one site*"*.

For part **(c)(i)** most candidates correctly answered 2, but then gave an incorrect description as to why the second row of the table in part **(ii)** was incorrect.

Parts **(d)** and **(e)** of the paper were concerned with the use of a file. For part **(d)**, some candidates were able to score 3 or 4 of the available marks. Candidates need to be familiar with the pseudocode keywords given in the syllabus for file handling which are given in ***section 2.2.3.***

For part **(e)(i)** the intention of this forerunner question to the writing of the program code for part **(ii)** was to ask candidates to state what variables would be required for the key parts of their coded solution.
The question proved to be a real discriminator. Often an Identifier and data type appeared to be worthy but was then contradicted by a description which did not relate to the problem. Candidates need much more programming experience, in particular in this application area of file handling as well as the suggested use of Boolean variables.

Reserved words often appeared as an Identifier - 'Date' being the most common suggested name for the date input by the user. This was penalised here in part **(e)(i)** but was not penalised a second time if used in the code for part **(ii)**. The program code solution seen suggests candidates need much more practical experience in writing file handling programs.

The Pre-release materials gave practical examples of the isolation of a sub-string from a string which was what was needed here for the isolation of the date and discount indicator from each line of the file.
A minority of candidates created some clever solutions seen for this task to extract the date and discount indicator.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

General comments

This was the first examination sitting for this new 9608 Computer Science syllabus.

For this Paper 2 candidates were expected to present for the examination having previously studied the Pre-release materials which had been circulated to centres. There were some excellent answers for the questions where the candidate was required to write program code in their chosen language, but there were a significant number of scripts where no attempt was made by the candidate for questions which required the writing of code and clearly such candidates were at a significant disadvantage. Candidates need extensive practical programming experience prior to sitting for the examination. There was little programming skill exhibited by a significant number of candidates.

The questions paper asked the candidate on page 2 to write the programming language they would be using (Visual Basic.NET, Python or Pascal). In the questions which required the writing of program code, the candidates was instructed to write the programming language used on the first dotted response line. Many candidates did not do this. The majority of candidates used Visual Basic, followed by Python, with a small minority using Pascal. Candidates need to be clear when reading the question if it is pseudocode or program code they are being asked to write. Some candidates stated 'pseudocode' as their programming language which did not gain credit.

The syllabus gives very detailed guidance on the key words to use when writing or completing a pseudocode algorithm. Candidates need to appreciate when it is appropriate to use the assignment operator or the '**=** ' symbol to test for equality.

Question 1

Part **(a)** was well answered with the majority of candidates gaining the full three marks. Candidates need to make sure that when they are asked to give an identifier name that it does read as such, with no <Space> characters. All too often the identifiers read like text – for example 'race time seconds' and not RaceTimeSeconds, or even worse 'RaceTime (Seconds) which suggests an array. This was an issue which carried through into later questions. Candidates need to follow some simple rules i.e. no space or inappropriate characters and adopt a consistent style such as CamelCase.

Part **(b)(i)** was well answered with a sensible name used for the personal best time. Data type INTEGER was expected. Some candidates had suggested in part **(a)**, the seconds input time be a real number (for example, time measured to the nearest hundredth of a second) and so declared the personal best time as REAL which gained credit.

For part **(b)(ii)** candidates gained the first two mark points for declaring variables and inputting the hours, minutes and seconds time. Most answers also followed this with the input of the personal best time. It was encouraging to see that candidates who wrote a Python solution did follow the question and include a comment statements giving the data type used for a variable.
Calculation of RaceTimeInSeconds was usually correct. Some candidates did not use the correct symbol for 'multiply' and did not gain credit.
Many answers omitted from their code the output of the RaceTimeInSeconds.
The logic for the last three mark points was widely misunderstood. It was a widespread error that if the race time was larger than the personal best time then this would general a new personal best time. Candidates need to be clear about the meaning of the <, >, and <> operators and their application.

For part **(c)(i)** the meaning of white-box testing was not well understood. The two key points looked for were "that the programmer would need to choose data values carefully" and "in order to test every path through the code and produce every possible outcome". Candidates needed to make it clear that the white-box tests carried out would assume knowledge of the particular code for this program. Some candidate answers read as if what was needed was the general skill of program coding.

Part **(c)(ii)** seemed to attract either very good or very weak responses.

The misunderstanding about whether or not the race time would or would not produce a new personal best carried through into this question about the selection of test data. Key points looked for were the selection of a dataset (hour, minutes and seconds) which was faster that the current personal best, followed by a dataset which gave a slower time, or vice versa. The final two marks were then for the display of the appropriate message. Candidates do need to be reasonably realistic with their data choice.
A significant number of candidates stated twice the same time values as given in the question on row one and the intention was unclear.

Question 2

For part **2(a)(i)**, most candidates received the mark for a statement about the input needing to be 'between 1 and 4'. The key wording in the question was 'what this pseudocode would do' and so a reference to "calling DisplayMenu" did not gain credit, whereas "displaying the menu choices to the user" scored.
Few candidates secured the second mark as they usually did not refer to the prompt.
Being able to clearly express an algorithm is a key skill for the computer scientist. This skill was again required later to secure the marks for **Question 5(b)(ii)**.

For part **(a)(ii)**, candidates should appreciate that the purpose of the algorithm was to 'validate' the input from the user.

Part **(b)(i)** was well answered. Most candidates correctly stated there would be three iterations of the loop.

For **part (b)(ii)**, most candidates secured the mark for stating that there would be a limit on the number of attempts allowed.

Part **(c)** was generally well answered. Common errors included the use of a different variable name – not Choice - and outputting the string "Call ReadFile" instead of pseudocode for a call to the ReadFile procedure. Answers using a Case structure were less likely to be correct than those using IFs. This was one of the questions where candidates did not gain any marks if they had incorrectly used the assignment operator ← instead of =.

Answers for the program code solution for part **(d)** were varied. It was good to see that candidates using Python did take note of the question and included a comment statement to state the data type used for their variables. However, some candidates read this as the cue to comment just about every line of their code, which was not the intention.
For Visual Basic and Pascal, the correct syntax was required for the declaration and assignment for constant i. Python candidates were only expected to show a normal assignment statement.
The mostly commonly gained marks were for:

- the declaration of one of the Integer variables
- the input of the choice value
- a correctly formed selection for the Choice value.

Code for the ReadFile and DisplayMenu procedures was often omitted and there were few solutions seen which had a correctly formed loop which included a call to DisplayMenu and which terminated with a user entry of 4.

Question 3

The majority of candidates scored well on part **(b)** but not for part **(a)**. Some candidates did not follow the question and wrote the module and data item descriptions on the dotted response lines (not the letter labels). A common error in part **(a)** was to label B as the control box.

Question 4

The majority of candidates achieved at least 3 marks for this question.

A common error was using the keyword NEXT instead of ENDFOR. Some answers confused the concept of a player 'throw' and a player 'total'. This misunderstanding was sometimes carried forward into the explanation for part **(ii)**. For part **(ii)**, candidates realised that if the both player totals were 10 then an inappropriate message was output.

Question 5

The syllabus distinguishes between a 1D and 2D array and so there was an expectation that the answers for part **(a)** should describe a 'ID array'. The second mark was awarded for a stated data type of integer.
The word 'array' was often mentioned, but frequently without the '1-D' prefix. 'Integer' was seen frequently.
There were many incorrect answers which included a specific pseudocode array declaration statement or mention of different structures (stack and queue).

The algorithm trace for part **(b)(i)** was generally well answered with a significant number of maximum marks.
Common errors included the omission of the final output of '3' and additional values present in the OUTPUT column. Less able candidates sometimes secured the mark for the values 1-7 in the DayNumber column, but then the first column values of 1, 2 and 3 (although correct) were placed on the incorrect row of the table and so did not gain credit. Candidates should be made aware that this horizontal alignment is significant when completing a trace table.

For part **(b)(ii)**, few candidates achieved the first mark point for the phrase 'first seven days'.
Most answers gained the second mark for some reference to total sales being 10 or more.
Output of the 'day' was often incorrectly stated in place of 'date' and output of the final total was often omitted. See the earlier general point made for **Question 2(a)(i)**.
For the table in part **(b)(ii)** candidates generally secured the mark for the first and third row. The second function call was often correctly identified as incorrect, but the explanation was insufficient and lacked any reference to 'parameters, for example, *"*only one character can be entered" or *"*can only look at sales from one site*"*.

For part **(c)(i)** most candidates correctly answered 2, but then gave an incorrect description as to why the second row of the table in part **(ii)** was incorrect.

Parts **(d)** and **(e)** of the paper were concerned with the use of a file. For part **(d)**, some candidates were able to score 3 or 4 of the available marks. Candidates need to be familiar with the pseudocode keywords given in the syllabus for file handling which are given in ***section 2.2.3.***

For part **(e)(i)** the intention of this forerunner question to the writing of the program code for part **(ii)** was to ask candidates to state what variables would be required for the key parts of their coded solution.
The question proved to be a real discriminator. Often an Identifier and data type appeared to be worthy but was then contradicted by a description which did not relate to the problem. Candidates need much more programming experience, in particular in this application area of file handling as well as the suggested use of Boolean variables.

Reserved words often appeared as an Identifier - 'Date' being the most common suggested name for the date input by the user. This was penalised here in part **(e)(i)** but was not penalised a second time if used in the code for part **(ii)**. The program code solution seen suggests candidates need much more practical experience in writing file handling programs.

The Pre-release materials gave practical examples of the isolation of a sub-string from a string which was what was needed here for the isolation of the date and discount indicator from each line of the file.
A minority of candidates created some clever solutions seen for this task to extract the date and discount indicator.

COMPUTER SCIENCE

Paper 9608/23
Written Paper

General comments

This was the first examination sitting for this new 9608 Computer Science syllabus.

For this Paper 2 candidates were expected to present for the examination having previously studied the Pre-release materials which had been circulated to centres. There were some excellent answers for the questions where the candidate was required to write program code in their chosen language, but there were a significant number of scripts where no attempt was made by the candidate for questions which required the writing of code and clearly such candidates were at a significant disadvantage. Candidates need extensive practical programming experience prior to sitting for the examination. There was little programming skill exhibited by a significant number of candidates.

The questions paper asked the candidate on page 2 to write the programming language they would be using (Visual Basic.NET, Python or Pascal). In the questions which required the writing of program code, the candidates was instructed to write the programming language used on the first dotted response line. Many candidates did not do this. The majority of candidates used Visual Basic, followed by Python, with a small minority using Pascal. Candidates need to be clear when reading the question if it is pseudocode or program code they are being asked to write. Some candidates stated 'pseudocode' as their programming language which did not gain credit.

The syllabus gives very detailed guidance on the key words to use when writing or completing a pseudocode algorithm. Candidates need to appreciate when it is appropriate to use the assignment operator or the '**=** ' symbol to test for equality.

Question 1

Part **(a)** was well answered with the majority of candidates gaining the full three marks. Candidates need to make sure that when they are asked to give an identifier name that it does read as such, with no <Space> characters. All too often the identifiers read like text, for example, 'number of previous wins' and not NumberOfPreviousWins, or an identifier which contained brackets which then suggests an array. Candidates need to follow some simple rules i.e. no space or inappropriate characters and adopt a consistent style such as CamelCase.

Part **(b)(i)** was well answered with the majority of candidates scoring the full three marks. Identifier names used followed closely the given structured English statements and data types of INTGER and STRING were used appropriately.
For part **(b)(i)**, the terms stepwise refinement or top-down design were not widely known.

The pseudocode algorithm for part **(b)(ii)** was well answered. Use of three separate IF statements or the use of nested IFs were the common answers for the selection. Some candidates used a CASE structure and gained full credit. Less able candidates often lost a mark for an incorrect syntax for the condition of 1 or 2 wins. These candidates often confused the use of the assignment and equals sign.

Question 2

Part **(a)** was intended to get the candidate familiar with the correct use of the MOD and DIV operators. Many answers scored the three available marks.

Answers for the completion of the test data table for part **(b)(i)** were varied. More able candidates were able to score the full three marks. The majority of candidates scored the mark for the Input value of 130 with a notes breakdown of 2-1-1 and the comment that this represents the 'least number of notes of each denomination'. Marks were often lost for the 85 value input when candidates completed the table with entries

of 1-1-1 and then gave a comment that the output was $5 short. The correct answer required the columns to be left blank with a comment that "This was not a multiple of $10". Similarly for the final row candidates often did not implement the question stating 'The maximum amount for a single withdrawn is $500'. Here, the three notes columns should have been left blank with a suitable comment.

Part **(b)(ii)** proved to be a good discriminator with all candidates able to score the first two marks for the input value Amount and the correct condition for Amount not being a multiple of $10.
There were various ways of arriving at the correct number of each note denomination. The most economical solution realised that the amount to be considered for each of the next two denominators was calculated by Amount MOD 50 and Temp MOD 20.

Question 3

Part **(i)** was well answered with many candidates gaining the five available marks. Some answers did not use the identifiers shown by the procedure parameters but instead gained credit for giving a description of the data item taken from the question.

For part **(ii)** few candidates gained the full five marks. Most candidates correctly realised that two of the parameters were the length and width, and of type integer. A common error was to state the data type for the job cost as integer. Few candidates understood the relevance of the length and width passed by value and the need for the job cost to be passed by reference. Candidates should be made aware that the syllabus has in detail the pseudocode syntax to be used for a procedure header in ***section 2.3.6***.

Question 4

Part **(a)** was intended as a forerunner to the use of the CONCAT function in the file handling tasks which followed. Most candidates realised that part **(i)** would generate an error as the second parameter was not a string. For parts **(ii)** and **(iii)**, candidates usually secured the two marks. Some candidates were not precise with the case of the strings, for example, the answer for part **(ii)** was stated with an upper-case P and in part **(ii)** the three words did not each start with the appropriate capital letter. These errors were penalised as accuracy is considered a key skill of the computer scientist.

Answers for part **(b)(i)** rarely scored the full six available marks. Candidates usually realised that DispenserCode was the parameter required for the CONCAT function and that the file name DISPENSERS was needed to complete the OPENFILE statement. Again candidates need to be prepared for a question like this by studying the given pseudocode syntax in ***Section 2.2.3*** of the syllabus.

In part **(ii)**, it was a requirement to secure the marks that the candidate referred to the data value which was to be input i.e. the dispenser code. Often candidates simply named a type of validation check, for example, length check and format check that did not gain credit without further expansion. Candidates need to be made aware that the key word in the question stem was to *'Describe …'*.

For part **(iii)**, candidates who had done some practical programming using file handling would have been aware that this scenario would cause the original file and its contents to be lost. Credit was given for the use of descriptors such as 'over-written' or 'deleted'. Often the key words used in the answer were vague, such as the original 15 records will be "covered up" or "covered over" which did not get credit.

For part **(iv)** the answer was 'append mode'. Some candidates answered with a program code statement using their chosen language which gained credit.

Part **(c)** required the candidate to write a program code solution. There should have been some straightforward marks available which did not require any knowledge of file handling. The stated requirement to output the total number of dispensers could have gained three marks for the initialisation of a variable, incrementing it and its final output. There were three available marks for statements which coded the basic file handling and input of the file, using the correct read/input mode and finally closing the file.
The isolation of the bank code and the dispenser code from the text line of the file was coded only by the most able candidates. Candidates need to be able to set up the loop for (in this case) the reading of each line from the file and the correct condition to terminate the loop. Candidates should be aware that some tasks will require the reading of every line from the file (as in this problem). Other tasks, for example the finding of a data value from a particular text line, will terminate the loop as soon as the value is found.

Question 5

It was rare to see two marks for responses for part **(a)(i)**. Key points expected were that a set of data values can be identified by a single identifier name. Credit was given if the candidate stated that the data items are usually of the same data type. Some candidates stated that each element is referenced using an index and that the index allowed direct access to each data item.

For part **(ii)**, most candidates correctly stated the value as 24.

For part **(iii)** the question required a clear statement describing what value was calculated. Despite the key word in the question stem *"Describe …"* some answers merely calculated the value of 38 with no explanation and so did not gain credit. Some candidates confused the worker number and day number.

For the completion of the trace table in part **(b)**, there were many answers which scored the full eight marks. A common error was omission of the zero entries resulting from the first FOR loop.

There were many varied answers given for part **(c)(i)**. Candidates generally scored two marks for the declaration of the worker number and day number as integer. The worker average was then often incorrectly declared with data type integer. Many candidates did not appreciate the significance in the question stem about the DailyHoursWorked and ProductionData and incorrectly declared these arrays as local variables.

Part **(ii)** was designed to replicate what often happens with practical code writing. The programmer may decide after the code is essentially working, that it would have been better designed as a procedure. Able candidates correctly identified the need to replace the FOR loop end values of 3 and 4. Only the very able candidates spotted the need to also replace the value 4 in the calculation of WorkerAverage.

For part **(iii)**, many correct answers were seen. Candidates need to be clear about the different context uses of the pseudocode keywords CALL and PROCEDURE.

COMPUTER SCIENCE

Paper 9608/31
Written Paper

General comments

This was the first Advanced Theory paper for this new syllabus.

Many candidates showed a limited knowledge of some of the new topics: encryption methods and usage, user-defined data types, bit manipulation and the BitTorrent protocol. It was encouraging to see many candidates display ability in using Boolean algebra to simplify a Boolean expression for a logic circuit.

There was evidence that, on occasions, candidates do not read and interpret the question as accurately as they should. Candidates often gave information that was not required. There were instances where candidates did not answer the actual question being asked, for example, see the comment on question **1(d)(i)** and the comment on question **3(b)(ii)**.

Comments on specific questions

Question 1

Recognising incorrect syntax proved easier than converting syntax diagrams to BNF notation.

(a) Marks were awarded provided there was some indication that the problem in part **(i)** was the lack of a semi-colon; that the problem in part **(ii)** was the presence of a '2'; and that the problem in part **(iii)** was the presence of an 'e'. In part **(iii)**, a number of candidates thought that only a single letter was permissible for a variable identifier and therefore stated, incorrectly, that both "dd" and "cce" were invalid.

(b) Fully correct answers were rare. Common errors consisted of enclosing the equals sign and the semi-colon in angle brackets in <assignmentstatement> and enclosing single characters in quotes in <letter> and <operator>. Despite the relatively greater complexity of <variable>, this was a better source of marks for many candidates, probably because it contained no terminal symbols.

(c) Despite the need for a recursive rule this was often answered correctly.

(d) (i) Many answers stated how interpreters operate but did not answer the actual question being asked: why do the programmers prefer to use the interpreter? Answers, which mentioned that it was easier but not easy to debug or that the code could be tested, even if incomplete, were acceptable answers.

(ii) Credit was given for indicating that the compiler produces a translated version of the source code and that it is this translated version that is distributed, not the source code.

Question 2

This question showed that candidates need more understanding of malware and encryption.

(a) Virus was a common, but incorrect answer to B. A virus attaches itself to another item of software; it is the worm which is a standalone piece of software.There was some confusion between pharming and phishing. Pharming is more than just the creation of a fake website.Pharming occurs when the victim is redirected to that fake website without their knowledge.Phishing starts with the victim receiving an electronic communication, typically an email. If the victim responds to the communication then it is likely to result in an attempt to elicit confidential information.

(b) The choice of A or B had to be clear. Some candidates failed to circle either A or B and consequently their answer had no value.The majority of answers chose option B. As the problems and solutions to viruses and worms are very similar and well-known, many candidates scored full marks. Some answers to spam stated that it was malware, but spam is an annoyance rather than a threat. Overloaded inboxes and the use of filtering would have gained full credit.

(c) This was not often answered well. There was a lack of clarity in the responses. Encryption is the conversion of plain text into cypher text, and is not the same as encoding. It was not sufficient to state that a public key is a key known by everybody ("public") or that it matches a private key. Some clarification over use of public key required. Either describing its use for encrypting a message that only the owner of the private key could decrypt or its use to decrypt a message that guarantees that the message came from the owner of the private key.

(d) (i) The majority of answers were equally split between digital signatures and digital certificates. Whereas a digital certificate is about authenticity of a public key, the digital signature is concerned with the authenticity of data which in this case is the software.

(ii) Candidates had limited knowledge of the process required to ensure that the software received is the same as the software sent. A number of candidates knew about the process at the sender's end: hashing the software to produce the message digest and the use of a private key to create the digital signature. Also what happens at the receiver's end: the use of a public key to decrypt the digital signature and then the checking that the message digests match.Incorrect statements were often made: the sender uses the public key of the receiver; the software is encrypted; the process involves parity. Even where candidates made the connection that their answer to part **(i)** was relevant in this part they usually did not describe the process that creates and uses the digital signature.

Question 3

Although candidates could select appropriate basic data types, there were difficulties with identifying and creating more structured data types.

(a) (i) Enumerated was rarely given as an answer. Date and string were common but incorrect answers.

(ii) A correct answer was rarely given. Some gave "structured" as a response but that was stated in the question and gained no credit.

(iii) Many incorrect answers gave a DECLARE statement rather than an assignment statement. There was little awareness shown of the use of the dot notation for accessing a field in a record. There were a number of answers where the statement given reversed the assignment so that MyMonthOfBirth was assigned to DateOfBirth.

(b) (i) Despite an example TYPE declaration being given in the question, some candidates wrote their own version of a declaration.These were usually not penalised if they exhibited an appropriate structure. Only a small number of answers successfully dealt with the need to use a structure which stored twelve values for the monthly rainfall totals. Consequently, four marks tended to be the most common score. Some candidates did not write a TYPE definition, but instead, their answer consisted of a discussion on the needs of the new data type. These answers scored no marks.

(ii) Many responses pointed out that the file would contain a small number of records or that the records would be in no particular order, but they did state how these facts would affect any searching of the file or the adding of new records to the file. Simply describing the properties of a serial file did not answer the question. The question posed a choice: serial organisation or sequential organisation? Credit was given where a statement demonstrated that serial organisation was more effective, for example, it is easier to add records to the end of a file rather than have to reorganise a file every time a new record is added. Some candidates used the word "file" when they were clearly writing about a record.

Question 4

This question was often answered well.

(a) Most candidates correctly completed the truth tables.

(b) (i) Appropriate Boolean expressions were usually given. There are a number of different ways of representing the logic operations AND, OR and NOT. Examination papers will use '.' for AND, '+' for OR and a bar for NOT. Although some candidates employed acceptable alternatives, it is hoped that candidates will use these conventions. Candidates also need to ensure that any Boolean expressions they write are clear regarding the scope of an operation, using brackets to remove any doubts.

(ii) Despite giving correct answers in part **(i)**, some candidates were not able togive the appropriate De Morgan's law.

(c) Candidates were mostly successful in giving the required Boolean expression. As in part **(b)(i)**, it is in candidates best interests to ensure that their expressions show clearly the scope of any operation.

(d) Only a minority of candidates could fully simplify the expression.However credit was given for any evidence of the correct application of De Morgan's laws or Boolean algebra.This resulted in many candidates scoring well on this question despite giving an incorrect or incomplete answer.

Question 5

Answers to this question were inadequate in justifying the need for particular items of hardware. Only a small number of candidates demonstrated knowledge of how to test and set bits.

(a) The majority of candidates correctly identified the system as a monitoring system.

(b) A temperature sensor was the most popular item. Other correct answers for items included a processor, a storage device and an analogue-to-digital convertor. Items not credited included: a timer (the processor software could sample the continuous sensor readings every hour); a monitor (the question specifically stated that the items were required for acquisition and recording); actuators (it is not a control system); and other types of sensor (only the temperature is to be recorded). Marks for justification were often very limited as candidates found it difficult to write answers that did little more than describe what the hardware items did: temperature sensors measured the temperature; analogue-to-digital convertors converted analogue signals to digital signals. Some answers were incorrect: the sensor does not record the temperature readings and the processor does not compare the temperature with some value. Marks were awarded for answers that added some accurate information about the required role of the hardware item: the temperature sensor is the input device for the system; the temperature sensor transmits temperature readings to an analogue-to-digital convertor; or a storage device stores the temperature readings from the sensor.

(c) (i) The question asked only for an interpretation of the contents of byte 1.The contents of byte 2 were not needed but this did not stop the majority of candidates giving a temperature reading. A number of answers stated that the 1 in bit 0 indicated that there was a location 0.

(ii) A number of answers set bit 0 of byte 1 to 1. Many candidates were successful in converting the integer -5 to a two's complement representation.

(d) A minority of candidates demonstrated a knowledge of how the use of AND and OR instructions could be used to test a bit or set a bit. Some candidates who gained no credit for the choice of operation did get credit for elements of an appropriate operand and/or the use of an immediate address mode.

Question 6

Candidates often displayed limited knowledge of the use of the BitTorrent protocol for file transfer.

(a) Many candidates believed that with three protocols, only three descriptions was required to be selected. The question clearly stated that “each description” should have a line to its appropriate protocol. Full marks were rare. The majority of candidates had little difficulty in choosing the correct protocol for the browser description. Choosing an appropriate protocol for the email actions proved much more problematical.

(b) Many candidates identified peer-to-peer as the correct answer. A number of candidates gave client-server as their response despite the hint in the question that client-server was not the answer.

(c) (i) Only a few responses demonstrated understanding of the role of the tracker in a torrent. Although BitTorrent uses peer-to-peer, the tracker is a server which aids communication between the peer computers. Many response stated that the tracker “tracks” or “keeps track of” "something" which was too vague to gain any credit.

(ii) Responses to this part were better than those in part **(i)** or part **(iii)**. Many candidates showed an understanding that seeds have a copy of the file and will upload parts of it on demand. The answer required recognition that the seeds were not servers but one of the peers in the peer-to-peer torrent.

(iii) It was rare to see a response that explained a swarm is all the connected users in a torrent. A few candidates did not explicitly state this fact, but conveyed the idea of “all the users” which gained credit.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

General comments

This was the first Advanced Theory paper for this new syllabus.

Many candidates showed a limited knowledge of some of the new topics: encryption methods and usage, user-defined data types, bit manipulation and the BitTorrent protocol. It was encouraging to see many candidates display ability in using Boolean algebra to simplify a Boolean expression for a logic circuit.

There was evidence that, on occasions, candidates do not read and interpret the question as accurately as they should. Candidates often gave information that was not required. There were instances where candidates did not answer the actual question being asked, for example, see the comment on question **1(d)(i)** and the comment on question **3(b)(ii)**.

Comments on specific questions

Question 1

Recognising incorrect syntax proved easier than converting syntax diagrams to BNF notation.

(a) Marks were awarded provided there was some indication that the problem in part **(i)** was the lack of a semi-colon; that the problem in part **(ii)** was the presence of a '2'; and that the problem in part **(iii)** was the presence of an 'e'. In part **(iii)**, a number of candidates thought that only a single letter was permissible for a variable identifier and therefore stated, incorrectly, that both "dd" and "cce" were invalid.

(b) Fully correct answers were rare. Common errors consisted of enclosing the equals sign and the semi-colon in angle brackets in <assignmentstatement> and enclosing single characters in quotes in <letter> and <operator>. Despite the relatively greater complexity of <variable>, this was a better source of marks for many candidates, probably because it contained no terminal symbols.

(c) Despite the need for a recursive rule this was often answered correctly.

(d) (i) Many answers stated how interpreters operate but did not answer the actual question being asked: why do the programmers prefer to use the interpreter? Answers, which mentioned that it was easier but not easy to debug or that the code could be tested, even if incomplete, were acceptable answers.

(ii) Credit was given for indicating that the compiler produces a translated version of the source code and that it is this translated version that is distributed, not the source code.

Question 2

This question showed that candidates need more understanding of malware and encryption.

(a) Virus was a common, but incorrect answer to B. A virus attaches itself to another item of software; it is the worm which is a standalone piece of software.There was some confusion between pharming and phishing. Pharming is more than just the creation of a fake website.Pharming occurs when the victim is redirected to that fake website without their knowledge.Phishing starts with the victim receiving an electronic communication, typically an email. If the victim responds to the communication then it is likely to result in an attempt to elicit confidential information.

(b) The choice of A or B had to be clear. Some candidates failed to circle either A or B and consequently their answer had no value.The majority of answers chose option B. As the problems and solutions to viruses and worms are very similar and well-known, many candidates scored full marks. Some answers to spam stated that it was malware, but spam is an annoyance rather than a threat. Overloaded inboxes and the use of filtering would have gained full credit.

(c) This was not often answered well. There was a lack of clarity in the responses. Encryption is the conversion of plain text into cypher text, and is not the same as encoding. It was not sufficient to state that a public key is a key known by everybody ("public") or that it matches a private key. Some clarification over use of public key required. Either describing its use for encrypting a message that only the owner of the private key could decrypt or its use to decrypt a message that guarantees that the message came from the owner of the private key.

(d) (i) The majority of answers were equally split between digital signatures and digital certificates. Whereas a digital certificate is about authenticity of a public key, the digital signature is concerned with the authenticity of data which in this case is the software.

(ii) Candidates had limited knowledge of the process required to ensure that the software received is the same as the software sent. A number of candidates knew about the process at the sender's end: hashing the software to produce the message digest and the use of a private key to create the digital signature. Also what happens at the receiver's end: the use of a public key to decrypt the digital signature and then the checking that the message digests match.Incorrect statements were often made: the sender uses the public key of the receiver; the software is encrypted; the process involves parity. Even where candidates made the connection that their answer to part **(i)** was relevant in this part they usually did not describe the process that creates and uses the digital signature.

Question 3

Although candidates could select appropriate basic data types, there were difficulties with identifying and creating more structured data types.

(a) (i) Enumerated was rarely given as an answer. Date and string were common but incorrect answers.

(ii) A correct answer was rarely given. Some gave "structured" as a response but that was stated in the question and gained no credit.

(iii) Many incorrect answers gave a DECLARE statement rather than an assignment statement. There was little awareness shown of the use of the dot notation for accessing a field in a record. There were a number of answers where the statement given reversed the assignment so that MyMonthOfBirth was assigned to DateOfBirth.

(b) (i) Despite an example TYPE declaration being given in the question, some candidates wrote their own version of a declaration.These were usually not penalised if they exhibited an appropriate structure. Only a small number of answers successfully dealt with the need to use a structure which stored twelve values for the monthly rainfall totals. Consequently, four marks tended to be the most common score. Some candidates did not write a TYPE definition, but instead, their answer consisted of a discussion on the needs of the new data type. These answers scored no marks.

(ii) Many responses pointed out that the file would contain a small number of records or that the records would be in no particular order, but they did state how these facts would affect any searching of the file or the adding of new records to the file. Simply describing the properties of a serial file did not answer the question. The question posed a choice: serial organisation or sequential organisation? Credit was given where a statement demonstrated that serial organisation was more effective, for example, it is easier to add records to the end of a file rather than have to reorganise a file every time a new record is added. Some candidates used the word "file" when they were clearly writing about a record.

Question 4

This question was often answered well.

(a) Most candidates correctly completed the truth tables.

(b) (i) Appropriate Boolean expressions were usually given. There are a number of different ways of representing the logic operations AND, OR and NOT. Examination papers will use '.' for AND, '+' for OR and a bar for NOT. Although some candidates employed acceptable alternatives, it is hoped that candidates will use these conventions. Candidates also need to ensure that any Boolean expressions they write are clear regarding the scope of an operation, using brackets to remove any doubts.

(ii) Despite giving correct answers in part **(i)**, some candidates were not able togive the appropriate De Morgan's law.

(c) Candidates were mostly successful in giving the required Boolean expression. As in part **(b)(i)**, it is in candidates best interests to ensure that their expressions show clearly the scope of any operation.

(d) Only a minority of candidates could fully simplify the expression.However credit was given for any evidence of the correct application of De Morgan's laws or Boolean algebra.This resulted in many candidates scoring well on this question despite giving an incorrect or incomplete answer.

Question 5

Answers to this question were inadequate in justifying the need for particular items of hardware. Only a small number of candidates demonstrated knowledge of how to test and set bits.

(a) The majority of candidates correctly identified the system as a monitoring system.

(b) A temperature sensor was the most popular item. Other correct answers for items included a processor, a storage device and an analogue-to-digital convertor. Items not credited included: a timer (the processor software could sample the continuous sensor readings every hour); a monitor (the question specifically stated that the items were required for acquisition and recording); actuators (it is not a control system); and other types of sensor (only the temperature is to be recorded). Marks for justification were often very limited as candidates found it difficult to write answers that did little more than describe what the hardware items did: temperature sensors measured the temperature; analogue-to-digital convertors converted analogue signals to digital signals. Some answers were incorrect: the sensor does not record the temperature readings and the processor does not compare the temperature with some value. Marks were awarded for answers that added some accurate information about the required role of the hardware item: the temperature sensor is the input device for the system; the temperature sensor transmits temperature readings to an analogue-to-digital convertor; or a storage device stores the temperature readings from the sensor.

(c) (i) The question asked only for an interpretation of the contents of byte 1.The contents of byte 2 were not needed but this did not stop the majority of candidates giving a temperature reading. A number of answers stated that the 1 in bit 0 indicated that there was a location 0.

(ii) A number of answers set bit 0 of byte 1 to 1. Many candidates were successful in converting the integer -5 to a two's complement representation.

(d) A minority of candidates demonstrated a knowledge of how the use of AND and OR instructions could be used to test a bit or set a bit. Some candidates who gained no credit for the choice of operation did get credit for elements of an appropriate operand and/or the use of an immediate address mode.

Question 6

Candidates often displayed limited knowledge of the use of the BitTorrent protocol for file transfer.

(a) Many candidates believed that with three protocols, only three descriptions was required to be selected. The question clearly stated that “each description” should have a line to its appropriate protocol. Full marks were rare. The majority of candidates had little difficulty in choosing the correct protocol for the browser description. Choosing an appropriate protocol for the email actions proved much more problematical.

(b) Many candidates identified peer-to-peer as the correct answer. Despite the obvious hint in the question that client-server was not the answer, there were a number of candidates who gave client-server as their response.

(c) (i) Only a few responses demonstrated understanding of the role of the tracker in a torrent. Although BitTorrent uses peer-to-peer, the tracker is a server which aids communication between the peer computers. Many response stated that the tracker “tracks” or “keeps track of” "something" which was too vague to gain any credit.

(ii) Responses to this part were better than those in part **(i)** or part **(iii)**. Many candidates showed an understanding that seeds have a copy of the file and will upload parts of it on demand. The answer required recognition that the seeds were not servers but one of the peers in the peer-to-peer torrent.

(iii) It was rare to see a response that explained a swarm is all the connected users in a torrent. A few candidates did not explicitly state this fact, but conveyed the idea of “all the users” which gained credit.

COMPUTER SCIENCE

Paper 9608/33

Written Paper

There were too few candidates for us to be able to produce a meaningful report.

COMPUTER SCIENCE

Paper 9608/41
Written Paper

Key messages

To succeed in this paper it is essential that candidates have practical experience of programming one of the following: Pascal, Visual Basic (console mode), and Python.

Programming and pseudocode questions from 9691 past papers and the tasks in the preliminary material for 9608/4 provide ideal topics for practical work.

General comments

There are many candidates who do not appear to have worked through the preliminary material. The responses to questions requiring the use of a programming language often showed a mismatch of stated programming language and the language used in the solution. Candidates need to be aware that they must produce program code in the language they declare at the beginning of the question part.

Candidates writing in pencil and then over-writing this in pen must ensure that their answers are legible. Pencil and pen show up the same to the marker.

Comments on specific questions

Question 1

This question covered the new topic of state transition diagrams. Most candidates were able to label the states and events correctly.

Question 2

Declarative programming has been a topic in the 9691 Paper 3 and should be well known.

(a) Most candidates were able to write the required facts using the syntax of the declarative programming language given in the question. Some candidates did not take note that predicates and atoms must be written starting with a lower case letter.

(b) Most candidates correctly stated that the result of the goal is ThisCity = manchester, london.

(c) Only the more able candidates were able to complete the rule correctly and write:

```
Countries_visited(ThisCountry)
IF city_visited(ThisCity) AND city_in_country(ThisCity, ThisCountry).
```

A common error was to write an imperative IF statement.

Question 3

Decision tables are a new topic. A question on this topic was included in the specimen paper 4.

(a) The majority of candidates were able to complete the first decision table. A common error was not to recognise that some combinations of the non-simplified table should result in no discount because the first row stated that goods did not total more than $20.

(b) The more able candidates were able to simplify the decision table by realising that the redundancies were where the condition 'goods totalling more than $20' was not met. No discount was applicable whatever the other conditions were. A common error was to give more than one action in a single column.

(c) This question part required candidates to write a function to return the discount amount. The question stated that the goods total and whether or not there was a discount card would be passed to the function as parameters. The more able candidates produced program code that followed the question requirements and made use of the simplified decision table. Nested IF statements were generally more successful than complex conditions involving AND. A suitable answer, given here in Python, would be:

```
def Discount(GoodsTotal, HasDiscountCard):
   if GoodsTotal <= 20:
      return 0
   else:
      if GoodsTotal > 100:
         if HasDiscountCard == True:
            return 10
         else:
            return 5
      else:
            if HasDiscountCard == TRUE:
               return 5
            else:
               return 0
```

Question 4

This question covered object-oriented programming which is not a new topic. A question was included in the specimen paper 4 and the topic was also covered in the preliminary material sent to Centres before the examination.

(a) Most candidates were able to draw linked boxes for the classes. Only the more able candidates had also added the arrow heads, correctly pointing from the subclass to the superclass. A common error was to show a class diagram instead of an inheritance diagram. Candidates should be aware of the different types of diagram used in object-oriented design.

(b) Some very good programming solutions were seen here. A common error was to omit declaring the properties as private and the methods as public.

(c) The mor able candidates realised that the method CalculatePay can be overwritten and there is no need for a different method for the subclasses. A common error was to list the properties and methods that were inherited from the superclass, rather than those that were required to distinguish each subclass from the superclass.

(d) The more able candidates stated that polymorphism allows the method CalculatePay to be declared in the superclass.

Question 5

This question covered sorting algorithms for an insertion sort and a bubble sort.

(a) (i) The more able candidates correctly completed the partial pseudocode algorithm of an insertion sort. A common error was to confuse the two pointers.

(ii) Many candidates correctly stated that the outer loop executes 9 times regardless of the order of the dataset and the WHILE loop is not entered at all because the condition is already false at the first encounter. A common error was to assume that the FOR loop iterated for anything other than 10 times.

(b) (i) The more able candidates realised that for each of the 9 iterations of the outer loop, the inner loop would iterate 9 times, making a total of 81 iterations.

(ii) The more able candidates were able to change the outer FOR loop to a REPEAT loop and introduce a Boolean variable, so that execution of the code finishes when no more swaps of adjacent values were made in a whole pass through the inner loop. A small number of candidates were able to rewrite the bubblesort algorithm to reduce the number of unnecessary comparisons in the inner loop. Candidates need to understand that, with each iteration of the outer loop, another value will arrive in its correct position. Therefore, the inner loop needs to compare fewer values each time. A suitably improved bubblesort algorithm is:

```
NumberOfItems ← 10
REPEAT
  NoMoreSwaps   ← TRUE

  FOR Pointer ← 1 TO NumberOfItems – 1
    IF NameList[Pointer] > NameList[Pointer + 1]
      THEN
        NoMoreSwaps ← FALSE
        Temp ← NameList[Pointer]
        NameList[Pointer] ← NameList[Pointer + 1]
        NameList[Pointer + 1] ← Temp
    ENDIF
  ENDFOR
  NumberOfItems ← NumberOfItems - 1
UNTIL NoMoreSwaps = TRUE
```

Common errors were to exclude one of the loops or to change the inner loop to a conditional loop.

Question 6

This question covered the queue abstract data type to be implemented as a linked list. This is not a new topic. The specimen paper 4 included a linked list question and the preliminary material also covered this topic.

(a) A small number of candidates showed the correct data values in the queue after the given operations were carried out. A common error was to treat the data as though it had entered a stack rather than a queue. The more able candidates linked the nodes correctly and inserted a null pointer in the last node. Only a few answers showed the inclusion of a head pointer and a tail pointer.

(b) (i) Many candidates correctly linked all the nodes. Some candidates inserted a null pointer in the last node. The free pointer often correctly pointed to the first node in the free list. The more able candidates also included null pointers for the head pointer and the tail pointer. A common error was to show a queue with existing data, rather than the required empty queue just after creation.

(ii) This question part gave the pseudocode for adding a data item to the queue. The more able candidates were able to complete the partial pseudocode for removing a data item from the queue. A common error was not to follow pointers to the next data item but to increment the pointer value by 1.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

To succeed in this paper it is essential that candidates have practical experience of programming one of the following: Pascal, Visual Basic (console mode), and Python.

Programming and pseudocode questions from 9691 past papers and the tasks in the preliminary material for 9608/4 provide ideal topics for practical work.

General comments

There are many candidates who do not appear to have worked through the preliminary material. The responses to questions requiring the use of a programming language often showed a mismatch of stated programming language and the language used in the solution. Candidates need to be aware that they must produce program code in the language they declare at the beginning of the question part.

Candidates writing in pencil and then over-writing this in pen must ensure that their answers are legible. Pencil and pen show up the same to the marker.

Comments on specific questions

Question 1

This question covered the new topic of state transition diagrams. Most candidates were able to label the states and events correctly.

Question 2

Declarative programming has been a topic in the 9691 Paper 3 and should be well known.

(a) Most candidates were able to write the required facts using the syntax of the declarative programming language given in the question. Some candidates did not take note that predicates and atoms must be written starting with a lower case letter.

(b) Most candidates correctly stated that the result of the goal is ThisCity = manchester, london.

(c) Only the more able candidates were able to complete the rule correctly and write:

```
Countries_visited(ThisCountry)
IF city_visited(ThisCity) AND city_in_country(ThisCity, ThisCountry).
```

A common error was to write an imperative IF statement.

Question 3

Decision tables are a new topic. A question on this topic was included in the specimen paper 4.

(a) The majority of candidates were able to complete the first decision table. A common error was not to recognise that some combinations of the non-simplified table should result in no discount because the first row stated that goods did not total more than $20.

(b) The more able candidates were able to simplify the decision table by realising that the redundancies were where the condition 'goods totalling more than $20' was not met. No discount was applicable whatever the other conditions were. A common error was to give more than one action in a single column.

(c) This question part required candidates to write a function to return the discount amount. The question stated that the goods total and whether or not there was a discount card would be passed to the function as parameters. The more able candidates produced program code that followed the question requirements and made use of the simplified decision table. Nested IF statements were generally more successful than complex conditions involving AND. A suitable answer, given here in Python, would be:

```
def Discount(GoodsTotal, HasDiscountCard):
   if GoodsTotal <= 20:
      return 0
   else:
      if GoodsTotal > 100:
         if HasDiscountCard == True:
            return 10
         else:
            return 5
      else:
            if HasDiscountCard == TRUE:
               return 5
            else:
               return 0
```

Question 4

This question covered object-oriented programming which is not a new topic. A question was included in the specimen paper 4 and the topic was also covered in the preliminary material sent to Centres before the examination.

(a) Most candidates were able to draw linked boxes for the classes. Only the more able candidates had also added the arrow heads, correctly pointing from the subclass to the superclass. A common error was to show a class diagram instead of an inheritance diagram. Candidates should be aware of the different types of diagram used in object-oriented design.

(b) Some very good programming solutions were seen here. A common error was to omit declaring the properties as private and the methods as public.

(c) The mor able candidates realised that the method CalculatePay can be overwritten and there is no need for a different method for the subclasses. A common error was to list the properties and methods that were inherited from the superclass, rather than those that were required to distinguish each subclass from the superclass.

(d) The more able candidates stated that polymorphism allows the method CalculatePay to be declared in the superclass.

Question 5

This question covered sorting algorithms for an insertion sort and a bubble sort.

(a) (i) The more able candidates correctly completed the partial pseudocode algorithm of an insertion sort. A common error was to confuse the two pointers.

(ii) Many candidates correctly stated that the outer loop executes 9 times regardless of the order of the dataset and the WHILE loop is not entered at all because the condition is already false at the first encounter. A common error was to assume that the FOR loop iterated for anything other than 10 times.

(b) (i) The more able candidates realised that for each of the 9 iterations of the outer loop, the inner loop would iterate 9 times, making a total of 81 iterations.

(ii) The more able candidates were able to change the outer FOR loop to a REPEAT loop and introduce a Boolean variable, so that execution of the code finishes when no more swaps of adjacent values were made in a whole pass through the inner loop. A small number of candidates were able to rewrite the bubblesort algorithm to reduce the number of unnecessary comparisons in the inner loop. Candidates need to understand that, with each iteration of the outer loop, another value will arrive in its correct position. Therefore, the inner loop needs to compare fewer values each time. A suitably improved bubblesort algorithm is:

```
NumberOfItems ← 10
REPEAT
  NoMoreSwaps   ← TRUE

  FOR Pointer ← 1 TO NumberOfItems – 1
    IF NameList[Pointer] > NameList[Pointer + 1]
      THEN
        NoMoreSwaps ← FALSE
        Temp ← NameList[Pointer]
        NameList[Pointer] ← NameList[Pointer + 1]
        NameList[Pointer + 1] ← Temp
    ENDIF
  ENDFOR
  NumberOfItems ← NumberOfItems - 1
UNTIL NoMoreSwaps = TRUE
```

Common errors were to exclude one of the loops or to change the inner loop to a conditional loop.

Question 6

This question covered the queue abstract data type to be implemented as a linked list. This is not a new topic. The specimen paper 4 included a linked list question and the preliminary material also covered this topic.

(a) A small number of candidates showed the correct data values in the queue after the given operations were carried out. A common error was to treat the data as though it had entered a stack rather than a queue. The more able candidates linked the nodes correctly and inserted a null pointer in the last node. Only a few answers showed the inclusion of a head pointer and a tail pointer.

(b) (i) Many candidates correctly linked all the nodes. Some candidates inserted a null pointer in the last node. The free pointer often correctly pointed to the first node in the free list. The more able candidates also included null pointers for the head pointer and the tail pointer. A common error was to show a queue with existing data, rather than the required empty queue just after creation.

(ii) This question part gave the pseudocode for adding a data item to the queue. The more able candidates were able to complete the partial pseudocode for removing a data item from the queue. A common error was not to follow pointers to the next data item but to increment the pointer value by 1.

COMPUTER SCIENCE

Paper 9608/43

Written Paper

There were too few candidates for us to be able to produce a meaningful report.