COMPUTER SCIENCE

Paper 9608/11
Written Paper

Key messages

Many of the candidates showed good paper management by presenting their work in a clear layout, writing their answer clearly, and showing where the answer continued elsewhere on the paper.

Candidates should be encouraged to write their answers clearly in the spaces provided on the examination paper. Candidates using additional sheets or blank spaces within the examination paper must indicate that the answer continues on the additional sheets. Candidates must also cross out earlier attempts or clearly indicate the answer to be marked.

It is important that candidates read the whole question carefully. Some candidates recognised certain 'key' words in some questions and wrote down everything they knew on that topic, rather than answering the question on the examination paper.

General comments

Overall there seemed to be a good understanding of the application of computers and the under-pinning logic. Candidates generally answered well the questions relating to logic and the transmission of data. The questions on bitmap graphics and the tasks of the operating system were more challenging.

Comments on specific questions

Question 1

The majority of candidates were able to link the communication media to the corresponding feature(s). The most common incorrect link was from "Copper cable" to "Least likely to have interference".

Question 2

(a) A minority of candidates answered this question correctly. Many need to improve their understanding of what is meant by a bitmap image. Most candidates were able to state that individual pixels made up the image, but only a few gave any further description.

(b) (i) This question was not answered well; many candidates need to improve their understanding of how a monochrome image is represented in binary and stored in a computer system.

(ii) Many of the answers to this question were generic descriptions of Run Length Encoding rather than a demonstration of how the image in the question would be encoded. When the question asks about a specific image, answers that describe repeating patterns, rather than repeated pixels are not sufficient for credit at this level.

(c) Many candidates found this calculation challenging. When a question asks candidates to show working, they must clearly show the different stages in the calculation. There were many missing stages for some solutions, so it was not possible to see how candidates had arrived at their answer. Some candidates need to improve their understanding of how many bits would be needed for a colour palette of 35 colours. As the question asked for an estimate, a divisor of 1000 to convert bytes to kilobytes was acceptable.

(d) Most candidates were able to state two benefits of vector graphics compared to a bitmap image. Many candidates did not give an appropriate reason for the benefit. Some candidates needed to be more precise with their answers and use the correct terminology, such as pixilation, rather than distortion.

Question 3

(a) Many candidates found all parts of this question challenging. The rubric named two operating system tasks and the question asked candidates to state the tasks performed. Many candidates did not read the question carefully and listed alternative operating system tasks to the ones given.

(i) There were a small number of very good answers to this part question. There was considerable confusion between memory and secondary storage, with the result that many candidates gave answers related to the management of secondary storage rather than the management of the computer memory. Many candidates need to improve their understanding of the difference between memory and secondary storage and ensure that they use the correct technical terminology in their answers.

(ii) There were a small number of very good answers to this part question. There was considerable confusion between the tasks performed by the operating system and the tasks performed by utility software.

(iii) This part of the question was answered better, with quite a few candidates able to state two other operating system tasks.

(b) The majority of candidates were able to identify the programs that were utility software.

Question 4

(a) Most candidates answered this question well. Candidates correctly identified the logic gates needed.

(b) Most candidates answered this question well, with candidates producing a correct truth table.

Question 5

(a) The majority of candidates were able to name two other servers. Some candidates need to read questions carefully. The question was set in the context of a college, and some of the answers given were not appropriate for this context.

(b) There were a small number of very good answers. In general, candidates did not answer this question well. Many candidates need to improve their understanding of the differences between the Internet and the World Wide Web.

(c) Most candidates found this question very challenging. Candidates need to improve their understanding of the sequence of events when a web page with embedded server-side code is requested. Candidates should also understand that, at this higher level, there is a need for precision and accuracy in their answers. For example, the browser software actually requests the desired web page from the server even though the student initiates the process.

Question 6

(a) The majority of candidates were able to identify correctly the missing parity bit. Some candidates confused odd for even parity and vice versa.

(b) (i) The majority of candidates answered this question well. A small number of candidates need to understand that it is the incorrect bit in the data block that is required, not the corresponding bit in the parity byte.

(ii) Many candidates found this part question challenging, even those who had correctly identified the incorrect bit in the previous part. Many candidates provided vague and imprecise answers. There was considerable confusion between the parity bit in each of the data bytes and the parity byte for the data block.

(c) A small minority of candidates were able to explain how there could still be errors in the data received yet these errors remain undetected using parity bits.

(d) Most candidates were able to name another method of data verification. Many found it more challenging to give a description of the method named. Candidates who gave a checksum as the method generally gave better descriptions. There is a need for candidates to improve their understanding of other methods of data verification during data transfer.

Question 7

(a) Almost all candidates were able to identify two input devices and one output device for the given system. Candidates must understand the need for precision in their answers; at this level, it is not enough to answer for example, remote. Candidates must use the full name for the device. There is also a need for careful reading of the question; the question asked for a “device”, so any type of media was incorrect.

(b) (i) Most candidates were able to state at least one difference between RAM and ROM. Many candidates need to be able to explain a difference. At this level of study, candidates need to give both sides of the comparison.

(ii) There were a small number of excellent answers to this question part, but many candidates need to read the question carefully. Answers had to relate to the context of the question, which was “use of RAM in games consoles”.

(iii) There were a small number of excellent answers to this question part, but many candidates need to read the question carefully. Answers had to relate to the context of the question, which was “use of ROM in games consoles”.

Question 8

(a) Many candidates were not able to explain the term register. The question states that registers are used in the Von Neumann model. A typical error was to describe how a register is constructed rather than explaining the term.

(b) (i) Many candidates found explaining the purpose of the MDR challenging. Candidates need to improve their understanding of the various registers used in the Von Neumann model of a computer system and how each individual register is used. A common error was to give the steps of the fetch-execute cycle in register notation, rather than to concentrate on the purpose of just the MDR.

(ii) Most candidates were able to name two other registers used in the fetch-execute cycle.

(c) (i) The majority of candidates correctly converted the binary value to denary.

(ii) The majority of candidates correctly converted BCD value to denary.

(iii) The majority of candidates correctly converted the two's complement binary value to denary. A small number of candidates need to understand the difference between unsigned binary and two's complement binary. A common error was the omission of the minus sign.

COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

Many of the candidates showed good paper management by presenting their work in a clear layout, writing their answer clearly, and showing where the answer continued elsewhere on the paper.

Candidates should be encouraged to write their answers clearly in the spaces provided on the examination paper. Candidates using additional sheets or blank spaces within the examination paper must indicate that the answer continues on the additional sheets. Candidates must also cross out earlier attempts or clearly indicate the answer to be marked.

It is important that candidates read the whole question carefully. Some candidates recognised certain 'key' words in some questions and wrote down everything they knew on that topic, rather than answering the question on the examination paper.

General comments

Overall, there seemed to be a good understanding of the application of computers and the under-pinning logic. Candidates generally answered well the questions relating to logic and the transmission of data. The questions on bitmap graphics and the tasks of the operating system were more challenging.
Candidates need to improve their understanding of scripting languages and the tasks of the operating system.

Comments on specific questions

Question 1

(a) Many candidates found both parts of this question challenging. The question asked candidates to expand on the two operating system tasks stated in the rubric. Many candidates did not read the question carefully and so listed alternative operating system tasks to the ones given, instead of answering the question on the examination paper. There was significant confusion between file management and memory management.

(i) The majority of candidates who answered in the context of file management were able to state one or more actions that can be performed on a file, such as open, close etc. However, candidates must understand that actions performed on a file is just one of the file management tasks performed by the operating system and so listing more than one such task is repetition.

(ii) There were a small number of very good answers to this part question. There was considerable confusion between the tasks performed by the operating system and the tasks performed by the device driver software or the printer firmware.

(b) (i) This part question was answered well. The majority of candidates were able to identify the applications that were utility software.

(ii) The majority of candidates were able to name two other utility programs. Candidates must take care to ensure that they write the names of the software correctly.

Question 2

(a) Many candidates gave completely correct answers to this question. Some candidates understood that the circuit for Y could be reduced to a single gate, even if the gate shown was often an incorrect NAND gate rather than the correct NOR gate.

(b) The majority of candidates completed the truth table correctly.

Question 3

(a) The majority of candidates were able to identify correctly the missing parity bit. Many candidates confused odd with even parity and vice versa.

(b) Candidates generally answered this question well. A small number of candidates need to understand that it is the incorrect bit in the data block that is required, not the corresponding bit in the parity byte.

(c) The majority of candidates understood that a format check and a range check were validation and that double entry was verification. Many candidates need to improve their understanding of the differences between a check digit and a checksum. Some candidates confused validation for verification and vice versa.

Question 4

(a) (i) The majority of candidates correctly converted the binary value to denary.

(ii) The majority of candidates correctly converted the binary value to hexadecimal.

(iii) The majority of candidates correctly converted the two's complement binary value to denary. A small number of candidates need to improve their understanding of unsigned binary and two's complement binary numbers. A common error was the omission of the minus sign.

(b) (i) Many candidates found it difficult to give a definition of a character. Candidates need to understand that a character set consists of more than just letters and numbers.

(ii) Most candidates were able to explain at least one difference between ASCII and Unicode. Candidates need to be able to give both side of the comparison when asked to explain differences.

(iii) Many candidates found this calculation challenging. When the question asks candidates to show working, it is necessary for them to show clearly the different stages in the calculation. There were many missing subscripts when values in different number bases were being added together, which often resulted in candidates giving an incorrect answer. For example, if 41_{16} was added to 25_{10}, without the subscripts the answer was given as 66.

Question 5

(a) There were a small number of excellent answers to this question. Many candidates need to improve their understanding of how analogue sound waves are converted into digital format. It is not enough at this level of study state, for example, "use an analogue to digital converter or the value is converted to digital form". Candidates must understand the need for precision in their answers, and should ensure, for example, that they make it clear that the time intervals are very small and regular.

(b) There were some good answers to this part question. Some candidates needed to read the question more carefully. The question asks candidates for the effects on the sound file in the stem of the question. Generic answers are not appropriate.

(c) There were some very good answers to this part question. Some candidates needed to read the question more carefully. The question asks candidates for the effects on the sound file in the stem of the question. There was some confusion between sample rate and sample resolution. Candidates must also understand that no credit can be given for repeating information given in the question.

(d) The majority of candidates were able to name two features of sound editing software. Many candidates need to understand the need for clarity in their explanations of purpose. At this level of advanced study, statements such as "the feature is cropping, and its purpose is to crop bits out of the file" are too imprecise. There needs to be some further explanation of the meaning of "cropping".

Question 6

(a) Most of the candidates were able to identify correctly the variable names. Some candidates need to understand that JavaScript is case sensitive, and that the precise names of the variables from the JavaScript code must be used.

(b) Almost all candidates were able to correctly identify the line of code that produced an output.

(c) Most candidates were able to state that the value was stored in or assigned to the variable `mark`. A minority of candidates correctly stated that the value came from a text box.

(d) (i) Many candidates correctly identified that the script would be run client-side. Some candidates need to improve their understanding of JavaScript as a client-side scripting language.

(ii) The majority of candidates did not answer this part well. Candidates need to improve their understanding of the differences between client-side and server-side scripting.

Question 7

(a) (i) The majority of candidates were able to correctly explain how the relationship between the two tables was implemented using the primary key / foreign key concept. Some candidates need to understand that in questions such as this it is necessary to name the attributes and the tables to which they belong.

(ii) Many candidates found drawing the entity-relationship (E-R) diagram from the given tables challenging. Most understood that there was a one-to-many relationship between the `USER` table and each of the other two tables. Often, the indication of the many end of the relationship was incorrect. A significant number of candidates also included an incorrect many-to-many relationship between the `PHOTO` and `TEXTPOST` tables.

(b) Many candidates found explaining referential integrity challenging. There was considerable confusion with data integrity. The better responses were those that used the given tables in their explanation. Many candidates need to improve their understanding of this topic.

(c) The majority of candidates did not answer this part well. The question states that the three tables are in 3NF and candidates are asked to define what is meant by each stage of database normalisation. There was considerable confusion between the process of normalisation and the definition of each stage, with the majority of candidates explaining the process of normalisation rather than what 1NF, 2NF and 3NF means.

(d) (i) There were a small number of very good answers to this part question. Many candidates need to improve their understanding of the use of basic SQL for the creation of database tables.

(ii) Similar to the previous part, there were a small number of very good answers to this part question. Many candidates need to improve their understanding of the use of basic SQL for the alteration of database tables.

COMPUTER SCIENCE

Paper 9608/13
Written Paper

Key messages

Many of the candidates showed good paper management by presenting their work in a clear layout, writing their answer clearly, and showing where the answer continued elsewhere on the paper.

Candidates should be encouraged to write their answers clearly in the spaces provided on the examination paper. Candidates using additional sheets or blank spaces within the examination paper must indicate that the answer continues on the additional sheets. Candidates must also cross out earlier attempts or clearly indicate the answer to be marked.

It is important that candidates read the whole question carefully. Some candidates recognised certain 'key' words in some questions and wrote down everything they knew on that topic, rather than answering the question on the examination paper.

General comments

Overall, candidates appear to have a good understanding of the application of computers and the under-pinning logic. Candidates generally answered well the questions relating to logic and the transmission of data. The questions on bitmap graphics and the tasks of the operating system were more challenging. Candidates need to improve their understanding of scripting languages and the tasks performed by the operating system.

Comments on specific questions

Question 1

(a) Most of the candidates correctly identified the variable names. Some candidates need to understand that the $ symbol is part of the variable name in PHP code, and that the exact identifier need to be used from the code need to be used.

(b) Some candidates were able to identify correctly the line of code that produced an output. A significant number of candidates need to improve their understanding of PHP code. A common error was to give line 22, the return statement, as the line number that produces an output.

(c) Most candidates were able to state that the value was used in the `calculateGrade` function. A minority of candidates correctly identified the value as a parameter to the routine. A small number of candidates correctly stated that the value came from a text box.

(d) Many candidates correctly identified that the script would be run server-side. Some candidates need to improve their understanding of PHP as a server-side scripting language.

Question 2

(a) Many candidates found drawing the entity-relationship (E-R) diagram from the given tables challenging. Most understood that there was a one-to-many relationship between the `PLAYER` table and each of the other two tables, but often the indication of the many end of the relationship was incorrect. A significant number of candidates also included an incorrect many-to-many relationship between the `LOGIN` and `PURCHASE` tables.

(b) Many candidates found explaining data integrity very challenging. The better answers were those that used examples from the given tables in their explanation, rather than a generic explanation. Many candidates need to improve their understanding of this topic.

(c) The majority of candidates were able to identify correctly that the statement given was true and that the database was in 3NF. A small number of candidates correctly stated two reasons why this was the case. Others need to improve their understanding of the meanings of First, Second and Third Normal Form.

(d) (i) There were a small number of very good answers to this part question. Many candidates need to improve their understanding of the use of basic SQL to create database tables.

(ii) There were a small number of very good answers to this part question. May candidates need to improve their understanding of the use of basic SQL to alter database tables.

Question 3

(a) The majority of students found describing the roles of ALU and CU very challenging. Candidates must understand that at this level of study it is too vague and imprecise to write statements such as “the ALU deals with arithmetic and logic, or the CU controls things”. This is simply re-writing information which has been given in the stem of the question.

(b) Many candidates need to improve their understanding of the different registers used in the Von Neumann model of a computer system. The best answers for the Status Register were those that gave examples of use. A common incorrect answer for the Program Counter was that it counted the number of instructions that had been fetched and executed.

(c) (i) The majority of candidates correctly converted the binary value into denary.

(ii) The majority of candidates correctly converted the binary value into hexadecimal.

(iii) The majority of candidates correctly converted the two’s complement binary value into denary. Some candidates need to understand the difference between unsigned binary and two’s complement binary. A common error was the omission of the minus sign.

(iv) This question was generally answered well. Some candidates need to improve their understanding of the limits of the values used in BCD. The most common incorrect answer was that the value in the first four bits exceeded 10, rather than 9.

Question 4

(a) Almost all the candidates were able to identify the missing parity bit. Some candidates confused odd for even parity and vice versa.

(b) A small number of candidates need to understand that it is the incorrect bit in the data block that is required, not the corresponding bit in the parity byte.

(c) The majority of candidates were able to identify correctly, whether the error detection measure was validation or verification. There was a small amount of confusion between validation and verification. The error detection method most often identified incorrectly was a checksum.

Question 5

(a) Most candidates correctly identified the two AND gates needed. A significant number of candidates did not recognise that an XOR gate was required with inputs C and D. The most common error was the use of an OR gate instead of the XOR gate.

(b) Many candidates completed the truth table correctly.

Question 6

(a) Many candidates need to improve their understanding of how a monochrome image is represented in binary. A common incorrect answer was 49, that is, the total number of bits required to represent the complete image rather than the number of bits needed for each pixel.

(b) Many of the answers to this question were generic descriptions of Run-length encoding rather than a demonstration of how the image in the question would be encoded with the given colour codes. When the question asks about a specific image, answers that describe repeating patterns, rather than repeated pixels are not sufficient for credit at this level.

(c) Candidates answered this part better than they answered part (a). Some candidates need to improve their understanding of how a 30-colour image would be represented in binary.

(d) There were some very good answers to this question. Many candidates need to improve their understanding of the contents of an image file header.

(e) The majority of candidates were able to name three features of graphics software that could be used to edit a photograph. Many candidates need to understand the need for clarity in their descriptions of the effects of the feature. At this advanced level of study, it is too vague and imprecise to write, for example, “the feature is cropping, and its effect is to crop bits out of the photograph”. There needs to be some further explanation of what cropping actually means.

Question 7

(a) There were some excellent answers to this question. Some candidates wrote full and detailed descriptions of the internal operation of different types of touchscreen. Some candidates needed to ensure that their responses are technically accurate and contain sufficient detail.

(b) (i) The majority of candidates were able to identify a suitable output device for the given system.

(ii) The majority of candidates were able to identify two suitable input devices for the given system. Many candidates need to ensure that they read the question carefully. The question asked for a statement about how each of the input devices would be used by the visitors. It is not appropriate to use vague and imprecise statements about the uses of the devices.

(c) The majority of candidates identified a suitable secondary storage device. Some candidates need to ensure that their answers are technically detailed enough for credit at this higher level of study. Answers that simply state a device is better, cheaper, or quicker are too vague and imprecise.

(d) There were a small number of excellent answers for this part. Many candidates need to read the question carefully. The question asks about the use of RAM and ROM in this computer system for the zoo. Answers must relate to this context.

COMPUTER SCIENCE

Paper 9608/21
Written Paper

Key messages

Candidates are expected to work through the pre-release material prior to the examination. This material includes a range of tasks designed to help candidates develop their problem-solving and programming skills. In addition, past papers give a clear indication of the types of question that candidates can expect.

There were some excellent programming solutions. It was clear that a significant number of candidates did not have sufficient practical programming experience prior to this examination.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and it is important that these are used correctly. It is also important that candidates use the correct syntax when writing or explaining algorithms using pseudocode. Candidates particularly need to appreciate when it is appropriate to use the assignment operator (←) as opposed to the equality operator (=).

General comments

If a candidate writes the answer to a question on an additional page or booklet, they must indicate where their revised answer is to be found. If answers have been crossed out, the new answers must be written clearly, so that Examiners can easily read the text and award the appropriate mark. Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful if this text is crossed out.

Visual Basic (console mode) and Python were the most popular languages, with only a very small minority using Pascal (console mode). As stated in the pre-release material, no marks were awarded for programming answers that did not use one of these three languages.

Candidates who offer solutions using Python need to take care to maintain the correct indentation, as this is important when defining the program structure. They must also remember to annotate the variables and their data types.

The following specific comments should be read in conjunction with the published mark scheme for this paper.

Comments on specific questions

Question 1

(a) Many candidates answered this question well. Identifier names containing spaces or non-meaningful identifier names such as '`ID`' and '`Name`' did not gain any credit as they were deemed too vague.

(b) (i) Candidates are reminded that they must make use of quotation marks (" ") to clarify between a literal string and an identifier. This was relevant to the first mark in this question where many candidates correctly identified the resultant string was "Month", but omitted the quotation marks.

(b) (ii) This question was generally answered well, with many candidates gaining all five marks. Some lost the second mark by stating String rather than a character for variable `MyInitial`.

Question 2

(a) (i) Many candidates were able to identify at least one feature in the code. Most candidates identified indentation as a feature. Few candidates gained all four marks.

(ii) There was a range of responses for this question with many providing the correct answers. Responses such as '#', did not receive credit. This is the symbol used before a comment statement in Python. If candidates give this symbol as an answer they must state what the symbol is used for.

(b) Candidates who have experience of programming would be able to answer many of the features asked for in this question. A common error was identifying line 3 as an example of an assignment, which is an example of declaration. Candidates need to recognise that an assignment will usually be identified by a back arrow symbol (←) in pseudocode.

Few candidates were able to correctly identify the number of function calls within the function. The most frequently seen incorrect answer was line 8 for the line of an unnecessary statement. This line initialises the string `OutString` to a value of "" (an empty string). This is essential for the first time the string is used in line 19, when a value is added to the existing string.

Question 3

Many candidates provided good responses for this question. Those that achieved zero or one mark appeared to be unfamiliar with the correct symbols used in a program flowchart. A program flowchart will almost always involve at least one decision box which is used to represent a conditional test. This could be for a conditional loop or a selection test (`IF ... THEN`). The decision box must always show two outcomes; True and False, and must go on to show the actions as a result of each condition. Many of the candidates who provided good responses, lost a mark for omitting the prompt to input the `SensorID`.

Question 4

(a) A minority of candidates provided correct responses for this part. It was evident from the quality of responses that candidates were unfamiliar with a structure chart.

Candidates often stated inputs and outputs as features shown on a structure chart. These are not a feature of a structure chart and it appeared that some candidates were mistaking a program flowchart for a structure chart.

(b) A significant number of candidates did not attempt this question. As with the previous question, candidates were not aware of how to draw a structure chart. The majority of candidates who attempted this question, made an attempt to draw another program flowchart which gained no credit. A minority of candidates were aware of the arrows used in a structure chart to show the direction of parameters but applied them incorrectly.

Question 5

(a) About half the candidates attempting this question correctly identified that a file will store the data after the program ends whereas an array only exists whilst the program runs and data is lost after its execution. A number of vague answers were given, such as easier/more efficient to add/store. These answers did not gain credit.

(b) Candidates who demonstrated good programming skills were more familiar with features of an IDE. The majority of the correct responses identified syntax checking, automatic indentation, and pretty printing.

(c) Python was the programming language used for the majority of the programming solutions. Many of the candidates remembered to include comments for variables and their data types. Candidates using Python must remember that indentation of code is important to identify the structure in their programs.

Candidates had to write a solution to add data to an existing file. They were expected to open the file in `APPEND` mode. The majority of solutions incorrectly used the `WRITE` mode. If the file is opened in `WRITE` mode, the existing data in the file will be overwritten. The question required that

the `MembershipNumber` be continually entered until it was equal to an empty string. Most responses correctly identified a need for some sort of loop. With the requirement to test `MembershipNumber` for an empty string, it was necessary for the variable to be declared as a string and not an integer.

Many candidates who made an attempt at this solution, achieved at least the mark for the input of the date of the scores.

Question 6

(a) (i) There were a variety of responses to this question part. Many incorrect answers referred to 1:5 being the ratio of data or 1 to 5 being the range of the data, rather than the range of index values. This question clearly split those candidates familiar with programming arrays and those with little experience.

The most popular correct answers gave reference to the size of the array or the range of indexes used in the array and the dimension of the array (1D). It was rare to see an answer which referred to the technical names of lower and upper bound.

(ii) Candidates who provided a response to this question generally gave the correct answer.

(b) A significant number of candidates did not attempt to answer this question. Many of these candidates provided some excellent solutions. Many of these candidates recognised the need for a nested loop to process a 2D array. Many candidates lost a mark for the incorrect use of the array index. In pseudocode, a 2D index is referenced with the syntax [x, y] rather than [x] [y]. Many candidates included a check to see if the new calculated value was greater than 255, but then did not limit the original element to 255. Some candidates made the mistake of changing a Boolean value to either True or False every time the new value was checked. This would result in an incorrect return value as it would return the result of the last value checked.

Question 7

As with the previous question, this question was attempted by around 75 per cent of the candidates with many excellent solutions and some very weak. Most of the weak solutions did not use any recognisable programming language.

Many candidates identified that a loop of 20 iterations was needed. Some of the Python programming solutions incorrectly used the range (0, 19) or (1, 20), which will only produce 19 iterations.

Most candidates recognised that variables for highest and total needed to be declared and initialised. Many responses used some elegant and inventive Python solutions, which did not require these variables. These solutions made use of built-in functions and methods such as sum and max. Full credit was given for their correct use.

Some solutions incorrectly returned the highest value rather than the position of the highest value.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

Key messages

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and it is important to use these correctly. It is also important that candidates use the correct syntax when writing or explaining algorithms using pseudocode. Some candidates used keywords or constructs that do not exist in their stated language.

Many candidates still seem to be unclear about the use of quotation marks to differentiate between an identifier name and a string. Some improvement in this area is noted but it is still a common problem.

Candidates need to read each question carefully before attempting to answer it. Different question papers may address topics in many different ways and the use of "stock" answers will not always be appropriate.

General comments

Although there were some excellent programming solutions, a significant number of candidates displayed low programming skills.

Visual Basic (console mode) and Python were equally popular languages, with only a very small minority using Pascal (console mode). As stated in the pre-release material, no marks were awarded for programming answers that did not use one of these three languages.
Python solutions were often the cleanest solutions, but there were also a number of excellent Visual Basic solutions.

Candidates who offered solutions using Python, need to take care to maintain the correct indentation, as this is important in defining the program structure.

If a candidate writes the answer to a question on an additional page or booklet, they must indicate where thet placed their revised answer. If answers have been crossed out, the new answers must be written clearly, so that Examiners can easily read the text and award the appropriate mark. Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful to cross out this text.

Comments on specific questions

The following specific comments should be read in conjunction with the published mark scheme for this paper.

Question 1

(a) **T**he majority of candidates answered this part well, indicating a good understanding of the need for meaningful identifier names.

A few names were considered too ambiguous and these included "`ID`" and "`Time`".

A small number of candidates gave data types instead of variable names.

(b) (i) Many candidates obtained full marks for this question. A common error was to omit the quotation marks for the first answer, which had to be used to differentiate between a string and an identifier name.

(ii) **T**he majority of candidates answered this part well. The most common error was the suggestion that variable "`Grade`' should be a string rather than a character. Solutions for alternative types to `REAL` for the `AverageMark` variable received credit.

Question 2

(a) (i) This part was not answered well. 'Comments' and 'White Space' were features correctly identified by many candidates, but many others either identified features of the code itself (e.g. "the use of loops"), or offered features that were not present in the example, such as 'indentation'.

(ii) Many candidates correctly stated 'Indentation', and a significant number referenced the use of appropriately named variables.

A significant number of candidates suggested features that would be provided by a PrettyPrint function within an IDE (such as 'colour differentiation'), rather than features of the source code.

(b) The majority of candidates answered this part well, indicating a good working knowledge of the basic structure and content of pseudocode. Common mistakes related to the identification of the final four features. '1' and '4' were often the incorrect answer for the number of local variables. Candidates classifying the input parameter as a local variable may explain '4': the thinking behind '1' is less obvious.

(c) (i) Many candidates correctly identified the return parameter mismatch and offered a suitable correction.

A common error was to say that variable `nc` (of type `Char`) was treated in the conditional statement (line 17) as an Integer. These candidates either missed or were unaware of the significance of the quotation marks surrounding the 0 and 9.

(ii) Candidates offered a wide range of responses to this question. The majority of candidates recognised that the conditional statement needed to be reversed, with the assignment coming after the '`THEN`' and the removal of the redundant '`ELSE`' clause.

Most solutions were based on an attempt to reverse the logic to end up with the following:

```
IF (nc >= '0') AND (nc <= '9')
   THEN
      c ← c + 1
ENDIF
```

Most candidates correctly changed the '`OR`' to '`AND`' but few of these correctly changed the conditional operators from the original '<' to the required '>='.

A significant number of candidates offered a solution based on inverting the original comparison but few of these included the additional set of brackets:

```
IF NOT ((nc < '0') OR (nc > '9'))
   THEN
      c ← c + 1
ENDIF
```

A minority of candidates suggested leaving the comparisons as they were, but instead changing the Boolean operator from `OR` to `NOR`, which received credit.

Question 3

Most candidates answered this question well. They demonstrated an understanding of the steps of the algorithm, and were able to express these using the program flowchart.

Where mistakes were made, they were usually in one of the following areas:

- Not indexing an individual element of the array
- Incorrect comparisons (use of '>=', rather than '>')
- Incorrect number of iterations

Although less common that in previous series, some candidates still attempted to combine operations of different types within a single symbol. Some candidates attempted to implement a loop structure within a *single* symbol, for example, `FOR Count ← 1 TO 100`

A minority of candidates did not use the correct concatenation in the output, but instead repeated the output example message given in the question.

Question 4

(a) (i) The majority of candidates did not answer this question well, and a significant number offered no answer.

A common error was the use of an unfilled circle at the start of the arrow, which indicated the return parameter, X. Several candidates drew the recursion arrow rather than the selection diamond.

(ii) The majority of candidates answered this question well.

Question 5

(a) This proved to be a challenge to many candidates. Marks were generally awarded for references to arrays being 'easier to search' or similar, and many references to the use of a subscript were seen. Some candidates stated a single identifier, less declaration and more efficient coding.

Common errors included references to arrays taking up less space in memory, as well as imprecise answers such as 'easy to access'.

(b) The most popular correct answer referred to dynamic syntax checking and this gained a mark for a minority off candidates.

Many candidates seemed to overlook the directive (in bold) and simply offered general features of an IDE. Similarly, a large number of candidates referred to later testing activities such as black and white-box testing.

(c) There were a small number of excellent full-mark answers. The majority of candidates did not gain more than two marks for this question part.

Most candidates presented a solution that contained the four required parts of the solution:

- Function heading
- Conditional loop structure
- Conditional statement
- Return statement

Most candidates gained the mark for the conditional loop structure. A significant number of candidates did not correctly declare the function in their chosen programming language. Many solutions included an input parameter that was not required.

The validation of the input number was often incorrect, with the constructs of the following type appearing very many times:

```
IF MembershipNumber >= 1111 AND <= 9999 THEN
```

The syntax for the return of the final string value was often incorrect, and many candidates attempted to output the value rather than return it.

Candidates need to take care when writing program code that they use the correct syntax for their chosen language. Using keywords or constructs that did not exist in the stated language was common. Many VB candidates used `Repeat ... Until` for a conditional loop, which is not valid syntax for that language.

(d) The second programming language question also had varied responses. There were some excellent full-mark answers. A significant number of candidates did not attempt this question.

Most candidates correctly identified the need for a loop used to read each line from the given file. Most candidates also attempted the final output statement. Some of these solutions used variables that had not been referenced previously and so did not gain a mark.

Many candidates decided to assign a value using an `INPUT` statement rather than from the use of the `GetNumber()` function. This may have been due to lack of practical experience of writing modular programs, or simply not reading the question carefully enough.

Although some improvement was evident, it appeared that a large number of candidates are still not familiar with the file handling syntax of their chosen language.

Question 6

(a) A minority of candidates answered this part correctly.

Common errors included 0/255, Min/Max, Highest/Lowest, First/Last.

Many candidates offered no answer.

(b) A number of candidates provided perfect answers. The method of swapping elements proved a challenge to many candidates. The majority of correct solutions followed the first example given in the mark scheme, but many also made use of a temporary row or a second 2D array.

Most candidates recognised the need for a nested loop, and included declaration statements for both loop counters and a temporary variable. Many candidates did not gain the first mark by not including an `END PROCEDURE` statement. As for previous questions, parameters that were not required were included in the procedure heading of many solutions.

A correctly nested `FOR ... NEXT` loop was the most common choice, although the count for the inner loop was often incorrect. Inner loops that ran from 1 to 8 would often flip the array twice, only to end with the original configuration. WHILE ... ENDWHILE loops were occasionally used, but these did not often work. The most common error was not re-initialising an index variable before the inner loop was started, resulting in an index value the started at 8, then decremented through 7, 6 and 5 (which was usually correct), but then continued to decrement indefinitely.

COMPUTER SCIENCE

Paper 9608/23
Written Paper

Key messages

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and it is important to use these correctly. It is also important that candidates use the correct syntax when writing or explaining algorithms using pseudocode. Some candidates used keywords or constructs that do not exist in their stated language.

Many candidates still seem to be unclear about the use of quotation marks to differentiate between an identifier name and a string. Some improvement in this area is noted but it is still a common problem.

Candidates need to read each question carefully before attempting to answer it. Different question papers may address topics in many different ways and the use of "stock" answers will not always be appropriate.

General comments

Although there were some excellent programming solutions, a significant number of candidates displayed low programming skills.

Visual Basic (console mode) and Python were equally popular languages, with only a very small minority using Pascal. As in previous sessions, no marks were awarded for programming answers that did not use one of these three languages. As before, Python solutions were often the cleanest solutions, but there were also a number of excellent Visual Basic answers. Pascal solutions were often very poor.

Candidates, who offered solutions using Python, need to take care to maintain the correct indentation, as this is important in defining the program structure.

If a candidate writes the answer to a question on an additional page or booklet, they must indicate where they placed their revised answer. If answers have been crossed out, the new answers must be written clearly, so that Examiners can easily read the text and award the appropriate mark. Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful to cross out this text.

Comments on specific questions

The following specific comments should be read in conjunction with the published mark scheme for this paper.

Question 1

(a) The majority of candidates answered this part well, indicating a good understanding of the need for meaningful identifier names.

Some candidates lost a mark for including spaces within the name and a few names were considered to be too ambiguous, particularly for the final item. Simply '`Weather`' was given on a few occasions but perhaps the most noteworthy of these was '`RainingOrNot`' which was suggested by a number of candidates. Although nothing in the question explicitly implies a Boolean data type, the meaning of the statement `RainingOrNot ← TRUE` is very ambiguous.

A small number of candidates gave data types instead of variable names.

(b) (i) Many candidates obtained full marks for this question. A common error was to omit the quotation marks for the first answer required to differentiate between a string and an identifier name.

(ii) The majority of candidates answered this part well. The suggestion that variable `Quality` should be a STRING rather than a CHAR was the most common mistakes. Sensible alternative types to REAL for the `Factor` variable were acceptable.

Question 2

(a) Few candidates gained the second mark for this part question. The majority of candidates gave a reasonable explanation for the use of 'Comments', but only a small minority explained that indentation helped the reader to understand the structure of the code. Some candidates confused indentation with initialisation.

(b) The majority of candidates answered this part well, indicating a good working knowledge of the basic structure and content of pseudocode. If mistakes were made, they were usually in identifying the final four features.

Common mistakes included:

- 2 or 100 as the number of dimensions of the array
- 1 as the number of times `OUTPUT` is called
- 1, 2 or 6 as the number of local variables

(c) (i) A small number of candidates correctly identified the return parameter mismatch and offered a suitable correction.

A number of candidates thought the error lay in the declaration of `DistinctionGrades` (line 42) rather than in the type mismatch caused in line 37.

(ii) There were a significant number of correct answers. Many candidates appeared to have only a vague working knowledge of the `CASE` construct. Most made some attempt at the initial '`CASE OF ThisMark`' statement and often the remaining clauses did map to the original nested `IF` statement, however in the majority of cases the subsequent 'syntax' was incorrect, even allowing for a broad interpretation of pseudocode. In many cases the only mark given was for the increment of variable `DGradeCount`.

Common mistakes included:

- The use of variable `Grade` rather than `ThisMark` in the `CASE` header
- Including the variable or the word `CASE` in each option statement
- Placing quotation marks around each option statement
- Omitting the `ENDCASE`

Question 3

(a) A minority of candidates correctly answered this question part. A wide selection of computing terms was suggested.

(b) The majority of candidates gained the first mark for representing the four modules in the correct relationship. Of these, the majority correctly added at least one set of interface parameters.

A small number of candidates offered no answer.

There were two common mistakes:

- Incorrect use of filled circles to indicate a Boolean parameter
- Parameters given in the reverse direction

Question 4

Most candidates demonstrated an understanding of the steps of the algorithm and were able to express these using the program flowchart. Some candidates did not know the correct symbols for the flowchart elements, whilst others ignored labelling of the branches from a decision box.

Where mistakes were made, they were usually in one of the following areas:

- Not indexing an individual element of the array
- Incorrect comparisons (use of '>=' rather than '>')
- Incorrect number of iterations
- Using `OUTPUT` rather that `RETURN`

Although less common that in previous series, some candidates still attempt to combine operations of different types within a single symbol. Some candidates attempted to implement a loop structure within a *single* symbol, for example: `FOR Count ← 1 TO 100`

Question 5

(a) (i) There appeared to be an equal number of correct and incorrect answers. Many candidates stated a three-dimension array, indicating a lack of understanding of array dimensions.

(ii) Candidates did not answer this part well, suggesting that they did not understand the reason a file is needed.
Many vague answers were seen, often referring to the data being ‘easy to find’ or similar.

(b) The most popular correct answer referred to the colour coding of keywords and this gained a mark for many candidates. Marks were sometimes lost due to candidates giving answers that were considered to be too brief. For example, ‘colour coding’ alone is insufficient.

The second mark was not given in most cases, with many candidates offering some form of error-detection feature.

(c) There were a small number of excellent full-mark answers. The majority of candidates only gained a mark or two.

Most candidates presented a solution that addressed the four required parts of the solution:

- Function declaration
- Loop structure
- Conditional statement
- Return statement

A significant number of candidates were unable to declare correctly the function in their chosen programming language. Many solutions included additional unrequired input parameters.

File handling statements were often incorrect.

Several candidates attempted to output the value for the average score rather than return it.

Candidates need to take care when writing program code that they use the correct syntax for their chosen language. Using keywords or constructs that did not exist in the stated language was very common. Many VB candidates used `Repeat ... Until` for a conditional loop which is not valid syntax for that language.

Question 6

(a) There appeared to be an equal split between correct and incorrect answers for this part question.

Common mistakes included '1', 'Integer' and 'variable'.

(b) Many candidates answered this part well. There were a number of perfect answers, usually following the solution given in the mark scheme.

Most candidates recognised the need for a nested loop and also included declaration statements for both loop counters and a Boolean variable. Many candidates lost the first mark by not including an `ENDFUNCTION` statement. As for previous questions, parameters that were not required were included in the procedure heading of many solutions.

A correctly nested `FOR ... NEXT` loop was the most common choice. Many candidates used `WHILE ... ENDWHILE` loops but often these did not work. The most common fault was not to re-initialise an index variable before the inner loop was started, resulting in an index value the started at 1, then incremented through 2, 3 and 4 (which was usually correct), but then continued to increment indefinitely.

A common mistake was to assign the Boolean variable a value in both cases (as below), meaning that the value returned represented only the case for the final pixel.

```
IF Picture[i, j] > MaxVal
   THEN
      Picture[i, j] ← MaxVal
      ClipFlag ← TRUE
   ELSE
      ClipFlag ← FALSE
ENDIF
```

Question 7

Many more candidates gained reasonable marks for this question than for question part **5(c)**, but a significant number only gained a mark or two.

Most candidates presented a solution that addressed the four required parts of the solution:

- Function declaration
- Conditional statements
- Return statement

A significant number of candidates were unable to declare correctly the function in their chosen programming language. A common mistake was to prompt for the input of the two integers rather than taking the values from parameters.

A significant number of candidates did not correctly use the MOD function (or equivalent) in their chosen program language. A small number attempted to implement this by using integer division and often this was correct.

COMPUTER SCIENCE

Paper 9608/31
Written Paper

Key messages

Candidates need to show an in-depth study of the topics and make good use of appropriate technical terminology on this paper. Those who have studied the theory and have practised the precise use of these tools and techniques were able to demonstrate successfully how they could use these to solve the problem set on the examination paper.

Questions that ask the candidate to 'Explain how…' requires a technical explanation of how to perform the task described in the question. Where questions ask for a benefit or drawback, responses must use the technology described in the question. Vague statements such as 'better', 'cheaper', and 'quicker' are not creditworthy.

General comments

Candidates need to read questions very carefully before attempting to write an answer. For example, in **Question 4(a)(i)**, the instruction is 'Write the Boolean expression for the truth table as a sum-of-products.' and in **Question 4(a)(iv)** the instruction is 'Write the simplified sum-of-products expression…'. These instructions require different answers. A simplified expression is not a suitable answer for **Question 4(a)(i)**.

Comments on specific questions

Question 1

(a) Many candidates found this part of the question challenging and did not understand that the number was negative. Some candidates achieved full marks.

(b) Most candidates answered this part well, with many responses scoring full marks.

(c) Most candidates correctly identified the effect on the range and the precision.

Question 2

(a) Many candidates found this part of the question challenging and could not clearly define the term **non-composite data type**.

(b) Most candidates could correctly identify non-composite data types; providing a description proved more challenging for some candidates.

(c) Many candidates found this part of the question challenging and could not clearly define the term **composite data type**.

(d) Most candidates could correctly identify composite data types; providing a description proved more challenging for some candidates.

Question 3

(a) (i) Many candidates stated a suitable benefit and a suitable drawback that applied specifically to a star network.

(ii) Many candidates stated a suitable benefit and a suitable drawback that applied specifically to a bus network.

(b) There was a full range of marks for this question.

(c) (i) Many candidates correctly identified the three layers.

(ii) Many candidates found this part of the question challenging and incorrectly described the use for BitTorrent. Some candidates gained full marks with many excellent explanations of how BitTorrent protocol allows file sharing.

Question 4

(a) (i) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks.

(ii) The majority of the responses showed a correctly completed Karnaugh Map.

(iii) The majority of the responses showed correct grouping.

(iv) Most responses showed a correct simplified sum-of-products.

(b) (i) The majority of responses showed a correctly completed Karnaugh Map.

(ii) Most responses showed correct grouping.

(iii) The majority of responses showed a correct simplified sum-of-products.

Question 5

(a) (i) The majority of candidates answered this part well.

(ii) The majority of candidates answered this part well.

(iii) The majority of candidates answered this part well.

(b) The full range of marks was seen. Responses that showed confidence in the use of Backus-Naur Form (BNF) notation provided fully correct answers. Many other responses did not show the correct use of the notation. A common error was the incorrect inclusion of round terminal symbols <>.

(c) There was a full range of marks for this part.

Question 6

(a) (i) Most candidates correctly identified the terms. Description for a given type of malware proved more of a challenge with many descriptions lacking the precision required.

(ii) Many responses correctly identified one solution; few candidates could clearly identify solutions to both threats.

(b) Some excellent responses correctly explained how a hashing algorithm could be used to ensure that the software was authentic and had not been altered.

Question 7

(a) Most candidates could correctly identify the type of system described.

(b) Many responses correctly identified two other items of hardware, and provided a justification that clearly related to the control system described in the question proved more challenging for most candidates.

(c) (i) The majority of candidates answered this part well.

(ii) The majority of candidates answered this part well.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

Key messages

Candidates must show an in-depth study of the topics and make good use of appropriate technical terminology for this advanced paper. Candidates, who have studied the theory and have also practised the precise use of these tools and techniques, were able to demonstrate successfully how they could be used to solve the problem set on the examination paper.

Questions that ask the candidate to 'Explain how….' require a technical explanation of how to perform the tasks described in the question.

General comments

Candidates need to read every question carefully before attempting to submit an answer. For example in **Question 5(c)** the instruction is 'State **three** vulnerabilities that a malware can exploit.' not state three possible effects of malware.

Comments on specific questions

Question 1

(a) Some candidates wrote correct pseudocode assignment statements. Other candidates found this part of the question challenging and made errors in the statement.

(b) (i) Many candidates correctly changed the upper bound of the array. Few candidates correctly showed the values.

(ii) Most candidates correctly rewrote the declaration.

(c) (i) Most candidates wrote pseudocode suitable for a composite data type. Some of the pseudocode was not entirely correct. Common errors included using incorrect data types for `Code` or `AverageMark`.

(ii) Some excellent responses to this question were seen, other candidates found this part of the question more challenging and could not explain how new data would be added to the file.

Question 2

(a) (i) The majority of candidates were able to provide a correct reason.

(ii) The majority of candidates were able to provide a correct reason.

(iii) The majority of candidates were able to provide a correct reason.

(b) The full range of marks was seen. Responses that showed confidence in the use of Backus-Naur Form (BNF) notation provided fully correct answers. Many other responses did not show the correct use of the notation.

(c) Most candidates drew a syntax diagram. Many candidates did not show the correct recursive structure for the changed variable.

Question 3

(a) Many candidates correctly calculated the normalised floating-point representation and showed their working.

(b) Some candidates found the correct denary value. Many candidates did not realise that both the mantissa and the exponent were negative.

Question 4

(a) Most responses gained some marks. A minority of responses were fully correct.

(b) Must candidates could identify at least one internet protocol; fewer candidates stated a correct use.

(c) Many candidates named the correct layer.

(d) (i) The full range of marks was seen with a variety of items in an IP data packet.

(ii) Few candidates could correctly identify and describe benefits of packet switching.

(iii) Few candidates could correctly identify items of data stored in a routing table.

Question 5

(a) Many candidates showed some understanding of the use of public and private keys. Few candidates provided the detailed technical explanation of how these keys were used to ensure that only the recipient of the email could read and understand the contents.

(b) The full range of marks was seen. Candidates that provided a detailed technical explanation of how Secure Socket Layer (SSL) or Transport Layer Security (TLS) helped to keep the confidential information secure gained full marks.

(c) Many candidates gained full marks for this question. Some candidates incorrectly stated the effects of malware rather than the vulnerabilities it can exploit.

Question 6

(a) Most candidates were able to link the scenario to the correct type of system.

(b) Many responses correctly identified two or three items of hardware. Most candidates found it more challenging to provide a justification that clearly related to the control system described in the question.

(c) (i) Most candidates were able to provide a correct answer.

(ii) Most candidates were able to provide a correct answer.

(d) Most candidates included correct load and store instructions. A common error was to use an `AND` op code instead of an `OR` op code.

COMPUTER SCIENCE

Paper 9608/33
Written Paper

Key messages

Candidates need to show an in-depth study of the topics and make good use of appropriate technical terminology on this paper. Those who have studied the theory and have practised the precise use of these tools and techniques were able to demonstrate successfully how they could use these to solve the problem set on the examination paper.

Questions that ask the candidate to 'Explain how…' requires a technical explanation of how to perform the task described in the question. Where questions ask for a benefit or drawback, responses must use the technology described in the question. Vague statements such as 'better', 'cheaper', and 'quicker' are not creditworthy.

General comments

Candidates need to read questions very carefully before attempting to write an answer. For example, in **Question 4(a)(i)**, the instruction is 'Write the Boolean expression for the truth table as a sum-of-products.' and in **Question 4(a)(iv)** the instruction is 'Write the simplified sum-of-products expression…'. These instructions require different answers. A simplified expression is not a suitable answer for **Question 4(a)(i)**.

Comments on specific questions

Question 1

(a) Many candidates found this part of the question challenging and did not understand that the number was negative. Some candidates achieved full marks.

(b) Most candidates answered this part well, with many responses scoring full marks.

(c) Most candidates correctly identified the effect on the range and the precision.

Question 2

(a) Many candidates found this part of the question challenging and could not clearly define the term **non-composite data type**.

(b) Most candidates could correctly identify non-composite data types; providing a description proved more challenging for some candidates.

(c) Many candidates found this part of the question challenging and could not clearly define the term **composite data type**.

(d) Most candidates could correctly identify composite data types; providing a description proved more challenging for some candidates.

Question 3

(a) (i) Many candidates stated a suitable benefit and a suitable drawback that applied specifically to a star network.

(ii) Many candidates stated a suitable benefit and a suitable drawback that applied specifically to a bus network.

(b) There was a full range of marks for this question.

(c) (i) Many candidates correctly identified the three layers.

(ii) Many candidates found this part of the question challenging and incorrectly described the use for BitTorrent. Some candidates gained full marks with many excellent explanations of how BitTorrent protocol allows file sharing.

Question 4

(a) (i) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks.

(ii) The majority of the responses showed a correctly completed Karnaugh Map.

(iii) The majority of the responses showed correct grouping.

(iv) Most responses showed a correct simplified sum-of-products.

(b) (i) The majority of responses showed a correctly completed Karnaugh Map.

(ii) Most responses showed correct grouping.

(iii) The majority of responses showed a correct simplified sum-of-products.

Question 5

(a) (i) The majority of candidates answered this part well.

(ii) The majority of candidates answered this part well.

(iii) The majority of candidates answered this part well.

(b) The full range of marks was seen. Responses that showed confidence in the use of Backus-Naur Form (BNF) notation provided fully correct answers. Many other responses did not show the correct use of the notation. A common error was the incorrect inclusion of round terminal symbols <>.

(c) There was a full range of marks for this part.

Question 6

(a) (i) Most candidates correctly identified the terms. Description for a given type of malware proved more of a challenge with many descriptions lacking the precision required.

(ii) Many responses correctly identified one solution; few candidates could clearly identify solutions to both threats.

(b) Some excellent responses correctly explained how a hashing algorithm could be used to ensure that the software was authentic and had not been altered.

Question 7

(a) Most candidates could correctly identify the type of system described.

(b) Many responses correctly identified two other items of hardware, and provided a justification that clearly related to the control system described in the question proved more challenging for most candidates.

(c) (i) The majority of candidates answered this part well.

(ii) The majority of candidates answered this part well.

COMPUTER SCIENCE

Paper 9608/41
Written Paper

Key messages

It is essential that candidates have practical experience of programming (including object-oriented programming) using one of the following languages: Pascal / Delphi (console mode), VB.NET (console mode) or Python. Programming and pseudocode questions from previous syllabus past papers and the tasks in the pre-release material provide some topics for practical work.

General comments

The majority of candidates demonstrated a good understanding of the basic principles across the paper. For example, they could write simple declarative language statements, and declare classes. Candidates found questions that required more than this basic overview, and needed application of knowledge to the given scenarios challenging.

Many candidates found the Jackson Structured Programming questions challenging. They found the creation of the PERT chart more straightforward. Candidates need to have more experience of producing object-oriented programs using their chosen programming language; this includes the language's conventions for constructors and encapsulation.

Comments on specific questions

Question 1

(a) Most candidates gained both marks. Common errors included missing the _ in `british_air`, and/or using capital letters that are not appropriate in these statements.

(b) Most candidates answered this part well. One common error was to use capital letters for the data returned, this needs to be the exact data used in the given declarative language code.

(c) This question was generally answered well. Some candidates put `glasgow` and `M` the wrong way around.

(d) Some candidates attempted to give a description that did not meet the requirements. The most common mark awarded was for an appropriate `AND`. Some candidates used the correct functions but had the variables in the incorrect places so they did not work.

(e) Most candidates answered this question well. Many candidates gave `True` as their answer, which is acceptable, but they should be aware that `YES` is the value that is actually returned.

Question 2

(a) There were mixed responses to this question. The most common correct elements were when the elements were swapped within the inner loop. Some candidates struggled to get the number of loop iterations correct, often putting 20.

(b) (i) Most candidates had a good attempt at this question. Candidates usually gained a mark for identifying that it keeps on looping. Fewer candidates explained why this is a problem, i.e. it will keep looping even though it is already sorted, that the extra iterations are unnecessary.

(b) (ii) Many candidates were able to describe how they would change the algorithm. Some of these responses were too vague; for example, they just described stopping the iterations when it was sorted, without the explanation of how they would actually do this. The most common marks were awarded for a description of using a Boolean flag to record if any comparisons had been made.

(c) Most candidates were able to identify at least one appropriate situation. Many candidates struggled to expand this; for example, a candidate would identify that a situation is when it is almost sorted, but would not be able to explain why.

Question 3

(a) Many candidates seemed unfamiliar with JSP. A flowchart was often drawn instead. Many candidates were able to identify the three boxes at level one, but struggled to expand further to `Operate Account`. Common errors included putting multiple boxes on one level. Many candidates did not attempt to put any selection of iteration within the diagram.

(b) Candidates were able to provide better answers to this part than for part (a). The first `Downloaded?` Level was commonly missed, but many candidates were able to identify the options of `Downloaded`, or `Not Downloaded` – although these rarely included the selection options. Candidates often followed through from these boxes and got the correct boxes on level four – again often without the selection symbol.

Question 4

(a) Most candidates answered this question well, giving the correct activities and durations.

(b) Many candidates struggled to identify the critical path. They instead worked out the longest time the tasks took to complete across the centre line, which were not always the critical path. Some candidates did not identify the tasks, instead they gave the timings, or node numbers, e.g. 1-2-3.

(c) Candidates struggled to identify the earliest start time. Most candidates gave the day that the previous task finished, e.g. 18. They did not identify that if that task finished at the end of day 18 then the next task would start on day 19. The latest finish time was more often correct.

Question 5

(a) There were mixed responses to this question. Most candidates were able to identify the attributes `Health`, `Strength` and `Direction`. Fewer candidates identified all of the methods, some candidates combined the different movements into one method which was acceptable, but fewer also included the constructor. Few candidates considered inheritance and often there was no attempt at the inheritance.

(b) Most candidates made a good attempt at this question and had clearly some experience of creating object-oriented programs in their chosen programming language. The class declaration was most often correct, along with the function declarations. More candidates than in previous years were able to write the correct constructor for their language. A common error within the constructor was assigning the parameter to the private variable, i.e. the assignment statement elements were on the wrong side of the statement. In the `GetDetails` function, some candidates took the values as parameters and then attempted to output these instead of accessing the object's attributes. Some candidates did not fully read the requirement for the function to return the values, instead outputting it, or attempting to return the values individually as separate statements which would not work because the function would end after the first return statement.

(c) As with part (c), responses to this part were mixed. A significant number of candidates were able to identify the inheritance in the class declaration. Fewer could implement this inheritance within the constructor, for example, not calling the parent class constructor with the relevant values. Most candidates had a good attempt at the function `GiveDamagePoints` and were able to check the correct values and return appropriate values. Some candidates did not use the attributes for the object, instead sending the values as parameters.

(d) (i) Candidates used a variety of methods to create a new object, for example by sending the values as parameters, or by creating a new object and then setting each value individually.

(ii) Candidates were often able to declare the function, but few could call the inherited method to get the string. Some candidates called the function but did this without taking the value it returned and using this in some manner, e.g. storing it in a variable, or concatenating it within the string. Some candidates outputted the string instead of returning it.

Question 6

(a) (i) Most candidates were able to define the appropriate type and both variables within it. Candidates need to ensure they read the question carefully. If the question asks for pseudocode, then the answer must be in pseudocode and not actual program code.

(ii) Many candidates were able to gain at least one mark, most commonly for declaring the array of type `ListNode`. Some candidates attempted to do this using their chosen programming language instead of pseudocode as the question required.

(b) There were mixed responses to this question. The final return value was most commonly correct, along with returning `Position`. Fewer candidates were able to write the correct code for the parameter value of `Scorer[Position].Pointer`.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

It is essential that candidates have practical experience of programming (including object-oriented programming) using one of the following languages: Pascal / Delphi (console mode), VB.NET (console mode) or Python. Programming and pseudocode questions from previous syllabus past papers and the tasks in the pre-release material provide some topics for practical work.

General comments

The majority of candidates attempted all questions. They demonstrated a good understanding of the basic principles.

Candidates need to have experience of developing and programming object-oriented solutions in one of the three stated programming languages.

Candidates must read all questions carefully and provide only the solutions required by the question.

Candidates need to have experience and knowledge of Jackson Structured Programming (JSP) and JSP structure diagrams.

Comments on specific questions

Question 1

(a) The minority of candidates were able to complete the algorithm using pseudocode. Candidates commonly filled the first space with a function `AddPhoto`, which is the name of the variable, used later, and therefore would not work. The final two variables were more commonly correct. Some candidates did not follow the capitalisation of the variables, and/or added spaces that then changed the name of the identifier.

(b) A significant number of candidates did not appear to have had enough experience of JSP diagrams. Some candidates missed out key sections, or gave inappropriate names to elements, for example, many candidates added a “Take Photo box” which was not part of this algorithm. When converting pseudocode to a JSP, candidates should follow the pseudocode carefully, ensuring they are using the terminology they have been given and that they are not adding additional features to the algorithm that are not required. Some candidates were able to place appropriate names in suitable places, but then missed the selection and iteration symbols, or put these in inappropriate places, for example, showing a repeated “Add a photo box” instead of it being an optional selection box.

Question 2

(a) This question was answered well by many candidates who were able to give both statements correctly. Common errors including capitalisation of key words e.g. `Gecko` or `Maxsize`. Many candidates put the `maxsize` of `gecko` to be `152` instead of `182`.

(b) Most candidates were able to gain at least one mark here. As with part (a), a common error was the use of capital letters, which are not appropriate, and some candidates did not include the underscore "_" in the names, e.g. `green_iguana`. It is important that candidates use the exact names given in the question.

(c) Most candidates provided the correct answer for this question. Some candidates put incorrectly reversed the positions of `squamata` and `X` within the brackets, or used capital `S` in `squamata`.

(d) A minority of candidates were able to provide a fully correct response. The most common mark awarded was for the use of an `AND` operator within their answer. Some candidates attempted to give a written description of the requirements as opposed to writing a line of code.

(e) This question was answered well with most candidates giving a suitable answer. A declarative language would return `YES` but many candidates gave `True` as a response, which was accepted on this occasion. Candidates should be aware of the correct terminology and how a declarative language returns its result.

Question 3

(a) Most candidates were able to answer this question and were able to identify that `CardData` was partially sorted. Some candidates stated incorrectly that `CardData` was fully sorted. There are still a couple of data items at the end that are not sorted.

(b) Some candidates identified the correct value for the number of iterations, but did not give further correct data items. Some candidates had the correct variables but wrote incorrect identifier names, e.g. included spaces that were not in the originals, or changed the case of some of the letters such as a lowercase `t` in `ValueToInsert`. It is important that candidates understand how identifiers are case sensitive and that `valuetoinsert` is not equivalent to `ValueToInsert`. Another common error was to give `HolePosition` on both occasions and not `HolePosition - 1`. It is important that candidates test run their completed algorithms to make sure they are working, as missing the `- 1` would stop the inner loop from working correctly.

(c) (i) A minority of candidates were able to explain why the search would not work. Many candidates described how a binary search works, and then stated that it must be in order otherwise it will not work; this was already given in the question. Candidates needed to think about why it did not work and explain this, for example, the item may be discarded too early, or the comparison would be invalid because the midpoint is not the numeric mid-point of the data values.

(ii) Most candidates were able to give a suitable description of a binary search within the given context. Some candidates wrote an algorithm, which did not answer the question. Some candidates gave a generic description of a binary search algorithm without any reference to the data as required by the question. Some candidates attempted to add together the data in the first and last elements, then dividing this by 2 to find the numeric midpoint between these values, instead of using the index of the first and last values.

(d) Most candidates were able to gain at least some of the marks. The most common correct answer was `CardData` in the first space. One common error was inaccurate, or lack of brackets in the calculation, for example `First + Last / 2` is not the same as `(First + Last) / 2`.

Question 4

(a) Many candidates were able to complete aspects of the class diagram. Most candidates were able to give the appropriate attributes for the official class. Fewer candidates gave the Team methods, many candidates attempted to identify an output method, e.g. `DisplayMember`, which was not given within the problem description. It is important that candidates carefully read the description of the problem to make sure they are identifying all of the required methods that are given within the context. Fewer candidates considered any inheritance between the classes, candidates were asked to complete the class diagram and they should be experienced in looking for inheritance as it is an important aspect and purpose of the diagram.

(b) Many candidates attempted to define the class using their chosen programming language. The most common marks were gained for declaring the class, the constructor and private attributes. Fewer candidates identified the need to pass the attributes to the constructor; this is best practice in a class design. Most candidates gave appropriate method declarations, and either output or returned the appropriate attributes.

(c) Many candidates were able to identify the need for inheritance. Fewer candidates were able to call the inherited constructor; in some cases, this was not attempted. Candidates should have experience of writing programs using inheritance, and inherited constructors.

(d) Candidates attempted a range of methods to declare an instance of an object within their programming language. Some candidates declared a variable named Official of type `BMXJudge`, and many did not declare it of the class type in their chosen language. Some candidates assigned the values as parameters; others set them direct (assuming the attributes were not private), and others made use of set methods, all of which were appropriate. A common error was not encompassing a string value in speech marks, e.g. assigning the `FirstName` the variable `Omar` instead of the string `'Omar'`. Some candidates did not include all of the data required by the question, e.g. they did not write the gender value to the object.

Question 5

(a) Most candidates were able to complete the GANTT chart accurately. Some candidates did not follow all of the dependencies on the table, for example, they showed task **G** as only dependent on **D** and not **E**.

(b)(i)(ii) Most candidates provided fully correct answers to these question parts.

(c) Most candidates answered this question part well. Most candidates were able to give at least one use of the GANTT chart by the manager, most commonly to find the critical path, or identifying which tasks can be in parallel.

Question 6

(a) Candidate provided mixed responses to this question part. Some candidates were able to follow the code and add the correct nodes in the correct places. Some candidates labelled the nodes in the order they were on the question paper, instead of the order that data items were added to the tree. Some candidates did not show how the nodes were connected, i.e. there were no lines, or pointers.

(b) Few candidates were able to give a complete solution to this question. Many candidates attempted to perform an in-order traversal, e.g. check left, output, and check right. This did not meet the requirement of the question, which only wanted the leaf nodes being output, i.e. when the `LeftPointer` and `RightPointer` are both `-1`. Some candidates used `IF` statements that would not fully work, for example they checked the `LeftPointer` then had the `RightPointer` comparison as an else. This would mean that if the `LeftPointer` were run then when the recursive call unwinds the `RightPointer` comparison would not run, which means that the function will not traverse to the right to find any additional leaf nodes. Many candidates did not check the pointer value in the binary tree because they did not identify `BinaryTree` as the array of nodes, and they attempted to access the pointer values in a variety of other ways that would not return the values required.

COMPUTER SCIENCE

Paper 9608/43
Written Paper

Key messages

It is essential that candidates have practical experience of programming (including object-oriented programming) using one of the following languages: Pascal / Delphi (console mode), VB.NET (console mode) or Python. Programming and pseudocode questions from previous syllabus past papers and the tasks in the pre-release material provide some topics for practical work.

General comments

The majority of candidates demonstrated a good understanding of the basic principles across the paper. For example, they could write simple declarative language statements, and declare classes. Candidates found questions that required more than this basic overview, and needed application of knowledge to the given scenarios challenging.

Many candidates found the Jackson Structured Programming questions challenging. They found the creation of the PERT chart more straightforward. Candidates need to have more experience of producing object-oriented programs using their chosen programming language; this includes the language's conventions for constructors and encapsulation.

Comments on specific questions

Question 1

(a) Most candidates gained both marks. Common errors included missing the _ in `british_air`, and/or using capital letters that are not appropriate in these statements.

(b) Most candidates answered this part well. One common error was to use capital letters for the data returned, this needs to be the exact data used in the given declarative language code.

(c) This question was generally answered well. Some candidates put `glasgow` and `M` the wrong way around.

(d) Some candidates attempted to give a description that did not meet the requirements. The most common mark awarded was for an appropriate `AND`. Some candidates used the correct functions but had the variables in the incorrect places so they did not work.

(e) Most candidates answered this question well. Many candidates gave `True` as their answer, which is acceptable, but they should be aware that `YES` is the value that is actually returned.

Question 2

(a) There were mixed responses to this question. The most common correct elements were when the elements were swapped within the inner loop. Some candidates struggled to get the number of loop iterations correct, often putting 20.

(b) (i) Most candidates had a good attempt at this question. Candidates usually gained a mark for identifying that it keeps on looping. Fewer candidates explained why this is a problem, i.e. it will keep looping even though it is already sorted, that the extra iterations are unnecessary.

(b) (ii) Many candidates were able to describe how they would change the algorithm. Some of these responses were too vague; for example, they just described stopping the iterations when it was sorted, without the explanation of how they would actually do this. The most common marks were awarded for a description of using a Boolean flag to record if any comparisons had been made.

(c) Most candidates were able to identify at least one appropriate situation. Many candidates struggled to expand this; for example, a candidate would identify that a situation is when it is almost sorted, but would not be able to explain why.

Question 3

(a) Many candidates seemed unfamiliar with JSP. A flowchart was often drawn instead. Many candidates were able to identify the three boxes at level one, but struggled to expand further to `Operate Account`. Common errors included putting multiple boxes on one level. Many candidates did not attempt to put any selection of iteration within the diagram.

(b) Candidates were able to provide better answers to this part than for part (a). The first `Downloaded?` Level was commonly missed, but many candidates were able to identify the options of `Downloaded`, or `Not Downloaded` – although these rarely included the selection options. Candidates often followed through from these boxes and got the correct boxes on level four – again often without the selection symbol.

Question 4

(a) Most candidates answered this question well, giving the correct activities and durations.

(b) Many candidates struggled to identify the critical path. They instead worked out the longest time the tasks took to complete across the centre line, which were not always the critical path. Some candidates did not identify the tasks, instead they gave the timings, or node numbers, e.g. 1-2-3.

(c) Candidates struggled to identify the earliest start time. Most candidates gave the day that the previous task finished, e.g. 18. They did not identify that if that task finished at the end of day 18 then the next task would start on day 19. The latest finish time was more often correct.

Question 5

(a) There were mixed responses to this question. Most candidates were able to identify the attributes `Health`, `Strength` and `Direction`. Fewer candidates identified all of the methods, some candidates combined the different movements into one method which was acceptable, but fewer also included the constructor. Few candidates considered inheritance and often there was no attempt at the inheritance.

(b) Most candidates made a good attempt at this question and had clearly some experience of creating object-oriented programs in their chosen programming language. The class declaration was most often correct, along with the function declarations. More candidates than in previous years were able to write the correct constructor for their language. A common error within the constructor was assigning the parameter to the private variable, i.e. the assignment statement elements were on the wrong side of the statement. In the `GetDetails` function, some candidates took the values as parameters and then attempted to output these instead of accessing the object's attributes. Some candidates did not fully read the requirement for the function to return the values, instead outputting it, or attempting to return the values individually as separate statements which would not work because the function would end after the first return statement.

(c) As with part (c), responses to this part were mixed. A significant number of candidates were able to identify the inheritance in the class declaration. Fewer could implement this inheritance within the constructor, for example, not calling the parent class constructor with the relevant values. Most candidates had a good attempt at the function `GiveDamagePoints` and were able to check the correct values and return appropriate values. Some candidates did not use the attributes for the object, instead sending the values as parameters.

(d) (i) Candidates used a variety of methods to create a new object, for example by sending the values as parameters, or by creating a new object and then setting each value individually.

(ii) Candidates were often able to declare the function, but few could call the inherited method to get the string. Some candidates called the function but did this without taking the value it returned and using this in some manner, e.g. storing it in a variable, or concatenating it within the string. Some candidates outputted the string instead of returning it.

Question 6

(a) (i) Most candidates were able to define the appropriate type and both variables within it. Candidates need to ensure they read the question carefully. If the question asks for pseudocode, then the answer must be in pseudocode and not actual program code.

(ii) Many candidates were able to gain at least one mark, most commonly for declaring the array of type `ListNode`. Some candidates attempted to do this using their chosen programming language instead of pseudocode as the question required.

(b) There were mixed responses to this question. The final return value was most commonly correct, along with returning `Position`. Fewer candidates were able to write the correct code for the parameter value of `Scorer[Position].Pointer`.