COMPUTER SCIENCE

Paper 9608/11
Written Paper

Key messages

This is a technical subject and candidates should use the technical language associated with the subject in an appropriate way.

Candidates must read each question carefully and use the context provided when answering the questions.

General comments

Most candidates were able to answer questions about sound representation and editing of sound files. Many candidates found the questions about video recording and low level programming much more challenging.

Comments on specific questions

Question 1

(a) There were some very good answers to this question with complete and detailed descriptions of the internal operation of various types of microphone. There appeared to be considerable confusion between the internal operation of the microphone and the conversion of the analogue sound waves to digital format.

(b) There were some excellent, detailed answers to this question. Candidates need to be aware that imprecise answers about decreasing the sampling resolution such as, '*it means that the sound takes up less space'* are not sufficient at this level.

(c) The majority of candidates were able to name two examples of sound editing software features. Some candidates found it more challenging to give a description of each feature. Candidates need to be aware that answers such as '*a feature of sound editing software is fade in/out which allows the sound to be faded in or out*' will not gain full credit. The description needs to add a little more than a repetition of the name of the feature.

(d) (i) Many candidates gave a good answer to this question. Candidates need to be aware that answers such as '*a frame rate of 60 fps means that 60 frames are displayed per second*' is not enough for credit. Candidates need to give some other description of a frame.

(ii) A minority of candidates were able to give a good description of progressive encoding. There was considerable confusion with interlaced encoding. Candidates need to improve their understanding of this topic.

(e) A minority of candidates understood that this was a file type that stored, for example, both audio and video data. Many candidates need to improve their understanding of this topic.

Question 2

(a) Most candidates were able to give a correct response for this question.

(b) (i) Most candidates found this question very challenging. The majority of candidates need to improve their understanding of the alternative valid formats of an IPv6 address.

(ii) There were many good answers to this question, with several completely correct solutions. There was some confusion about IP addresses assigned by the Internet Service Provider. This was the row where the tick was most likely to be in the wrong column.

(c) Many candidates were able to give two types of transmission media other than copper cable. Many candidates need to understand that coaxial cable and twisted pair cable are both types of copper cable.

Question 3

(a) The majority of candidates were able to describe open source and shareware.

(b) The question asked for a justification of the licence Hugo should choose. There were a small number of good answers to this part question. A number of candidates repeated a description given in part **(a)** of one of the licence types, which did not answer the question.

Question 4

(a) (i) Many candidates found this part question challenging. Candidates need to improve their understanding of direct addressing and indirect addressing.

(ii) Some candidates correctly explained how the `ADD 20` instruction could be interpreted as either direct or indirect addressing. A significant number of candidates need to improve their understanding of these addressing modes. The most common error was the suggestion that using direct addressing the denary value 20 would be loaded.

(b) Many candidates correctly identified the instructions that used absolute and symbolic addresses. The incorrect answer seen most often was a complete reversal of the correct answer.

(c) (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) Many candidates correctly converted the binary value to denary. Some candidates found this more challenging. A frequent error was the omission of the minus sign.

(d) The majority of candidates were successful in completing the first line of the trace table. Many candidates found completing the rest of the table challenging. There were a small number of completely correct answers.

Question 5

Candidates needed to justify whether the actions of the individuals in each of the three scenarios were ethical or unethical. Candidates must be aware that justifying an action requires evidence/argument for the action. The majority of candidates found this challenging. Many candidates gave descriptions of alternative actions or consequences of the actions that did not answer the question. Stronger responses related the actions of the individuals in each scenario to one or more of the eight principles listed in the ACM/IEEE Software Engineering Code of Ethics.

Question 6

(a) Many candidates correctly answered this question. Some candidates need to understand that PHP code is case sensitive and that the $ symbol is part of the variable name. Many candidates omitted the $ symbol.

(b) The majority of candidates correctly answered this question.

(c) Many candidates found this question challenging. Candidates need to improve their understanding of the purpose of a `return` statement.

(d) Many candidates found this question challenging. There were a small number of good answers. In general, responses needed to be more precise and accurate. Candidates need to improve their understanding of the interaction between the client browser and the web server, when the browser requests a web page containing a server-side script.

Question 7

(a) Most candidates answered this question well. The most common error was the incorrect indication of the many ends of the one-to-many relationships.

(b) Candidates needed to explain how the primary and foreign keys are used to link the tables in the **movie theatre database**. Many candidates gave generic answers with very little or no reference to the database mentioned in the question. It is essential that candidates carefully read the question and give a response in the given context.

(c) Most candidates answered this question well. The most common error was the omission of the word `TABLE` in the `ALTER TABLE` clause, and the omission of a data type in the `ADD` clause.

(d) A significant number of candidates were able to write the correct SQL script. The most common reason for not gaining all the marks was the omission of the second table, usually `MOVIESCHEDULE` in the `FROM` clause.

COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

This is a technical subject and candidates should use the technical language associated with the subject in an appropriate way.

Candidates must read each question carefully and use the context provided when answering the questions.

General comments

Most candidates were able to answer questions about cabling and hardware. Many candidates found the questions about low-level programming much more challenging.

Comments on specific questions

Question 1

(a) (i) Many candidates need to understand that the number of bits, x, required for an n-colour bitmap is given by $n = 2^x$. The most common incorrect answer was 4 bits.

(ii) The majority of candidates correctly calculated the number of pixels and the minimum file size of the image. Some candidates showed some confusion between the terms bits and bytes.

(b) (i) The majority of candidates correctly calculated the number of pixels in the image. Some candidates showed confusion between bits and bytes. Some candidates need to understand the relationship between bytes and megabytes.

(ii) The majority of candidates were able to match the method of compression to the appropriate description and compression type.

(c) Candidates needed to read the question carefully and give responses in the context. Responses should have related to the use of the logo on the website, as the question stated that the logo was for the company website. Candidates need to provide more precision in their response. A response such as '*a vector graphic uses less space*' lacks detail and precision.

Question 2

(a) The majority of candidates were able to identify the correct sequence of steps.

(b) Many candidates gave good descriptions of the purpose of an IP address. Some candidates need to be aware that it is the device that is identified not the user.

(c) (i) Most candidates were able to give two benefits of fibre-optic cable. Some candidates found it more challenging to give three benefits.

(ii) Some candidates found giving drawbacks of fibre-optic cable more challenging. Answers such as '*fibre-optic cable is more expensive'* lack detail and are imprecise.

Question 3

(a) (i) Many candidates found this part question challenging. Candidates need to improve their understanding of symbolic addressing and absolute addressing.

(ii) Some candidates gave correct examples of an ADD instruction using absolute addressing. A minority of candidates were able to give a correct example of a similar instruction using symbolic addressing.

(b) (i) Many candidates found this part question challenging. Candidates need to improve their understanding of indexed addressing and immediate addressing.

(ii) A small number of candidates were able to give an example of an instruction using indexed addressing. A minority of candidates gave a correct example of immediate addressing.

(c) (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) Many candidates correctly converted the binary value to denary. Some candidates found this more challenging. A frequent error was the omission of the minus sign.

(d) The majority of the candidates were successful in completing the first line of the trace table. Many candidates found it challenging to complete the rest of the table. There were a small number of completely correct answers.

Question 4

(a) Some candidates need to improve their understanding of register notation. Common errors included the incorrect use of brackets and assignment symbols often pointed in the wrong direction.

(b) (i) Many candidates gave three good examples of events that could generate an interrupt. Some candidates need to ensure that their answers are precise.

(ii) The majority of candidates were able to state some relevant points. A small number of candidates gave good descriptions of how the processor handles interrupts.

(c) The majority of candidates correctly linked the buses to the correct descriptions.

Question 5

Candidates were required to justify whether the actions of the individuals in each scenario were ethical or unethical in three scenarios. Candidates need to be aware that justifying an action requires evidence/argument for the action. The majority of candidates found this challenging. Many candidates gave descriptions of alternative actions or consequences of the actions that did not answer the question. Stronger responses related the actions of the individuals in each scenario to one or more of the eight principles listed in the ACM/IEEE Software Engineering Code of Ethics.

Question 6

(a) Candidates needed to read this question carefully. The question asked for two **electronic** measures. Many candidates gave at least one method that was not electronic.

(b) (i) Many candidates found it challenging to describe what was meant by a library routine. Some candidates appreciated that the code would be pre-written. Few were able to give a second relevant point.

(ii) More candidates were able to describe a benefit of using library routines. Fewer candidates were able to describe a drawback. Candidates must be aware that statements such as, '*the library routine may contain a virus*' are too incomplete for a description at this level.

(c) Candidates needed to read this question carefully. The question asked why a program developer would need to use **both** an interpreter and a compiler. Quite a few candidates gave generic descriptions of the two programs that did not answer the question.

COMPUTER SCIENCE

Paper 9608/13
Written Paper

Key messages

This is a technical subject and candidates should use the technical language associated with the subject in an appropriate way.

Candidates must read each question carefully and use the context provided when answering the questions.

General comments

Most candidates were able to answer questions about cabling and hardware. Many candidates found the questions on low-level programming much more challenging.

Comments on specific questions

Question 1

(a) (i) Many candidates need to understand that the number of bits, x, required for an n-colour bitmap is given by $n = 2^x$. The most common incorrect answer was 6 bits.

(ii) The majority of candidates correctly calculated the number of pixels and the minimum file size of the image. Some candidates showed some confusion between bits and bytes.

(b) (i) The majority of candidates correctly calculated the number of pixels in the image. There was some confusion between bits and bytes. Some candidates need to understand the relationship between bytes and gigabytes.

(ii) The majority of candidates correctly identified whether the compression method was lossy or lossless.

(c) Most candidates were able to include an example of how at least one row of the given image would be compressed using run-length encoding (RLE). The question asked how run-length encoding would compress the **image** and answers should have reflected this. Many candidates found this challenging.

Question 2

(a) Many candidates found this part question challenging. Candidates need to improve their understanding of relative addressing and indexed addressing.

(b) (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) Many candidates correctly converted the binary value to denary. Some candidates found this more challenging. A frequent error was the omission of the minus sign.

(iv) The majority of candidates were able to show the result of the general purpose register after the instruction was run.

(c) The majority of candidates were successful in completing the first line of the trace table. Many candidates found it challenging to complete the rest of the table. There were a small number of completely correct answers.

Question 3

Candidates were required to justify whether the actions of the individuals in each of the three scenarios were ethical or unethical. Candidates must be aware that justifying an action requires evidence/argument for the action. The majority of candidates found this challenging. Many candidates gave descriptions of alternative actions or consequences of the actions that did not answer the question. Stronger responses related the actions of the individuals in each scenario to one or more of the eight principles listed in the ACM/IEEE Software Engineering Code of Ethics.

Question 4

(a) The majority of candidates successfully identified the missing steps in the sequence.

(b) (i) Most candidates were able to identify two reasons why the IPv4 address given was invalid.

(ii) The majority of candidates need to improve their understanding of IPv6 addresses. Some candidates understood that L was an invalid hexadecimal digit. Only a minority of candidates were able to give a second reason why the address given was invalid.

(c) There were a number of completely correct responses to this question. The last row was the most likely to be incorrectly identified.

Question 5

(a) (i) This question was well answered. Many candidates gave good complete descriptions of a commercial software licence.

(ii) **T**he most popular choices of alternative software licences were either open source or shareware, and there were many detailed descriptions of both of these.

(b) The question asked for three ways that users' details are kept secure in the context of a software developer releasing a game over the Internet. Candidates need to read the question carefully. If the game is released on the Internet, then isolating the computer from any public network will not be an option for ensuring the security of user data.

Question 6

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

(ii) Many candidates found this part challenging. They need to improve their understanding of the use of brackets in register notation.

(b) The majority of candidates successfully stated the meaning of the term interrupt.

(c) The majority of candidates answered this question well, with many giving completely correct responses.

Question 7

(a) Many candidates provided good responses to this question. Some candidates need to improve their understanding of this topic. Candidates also need to understand that three answers related to a single benefit, for example, cost, only gain credit once. Imprecise answers such as '*a wireless network is cheaper*' are also not enough at this level.

(b) Some candidates found it more challenging to give a drawback of a wireless network. Candidates need to understand that imprecise answers related to, for example, cost or efficiency would not be sufficient at this level.

Question 8

Most candidates answered this question well, with many providing completely correct solutions to both parts. Some candidates need to be aware of the importance of drawing logic gate symbols accurately.

COMPUTER SCIENCE

Paper 9608/21
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and be able to apply them to the scenarios presented if they are to achieve high marks.

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 important that candidates writing program code use the correct syntax for their chosen language and they must show a good understanding of fundamental programming concepts. Examples include the difference between a literal and an identifier as well as the difference between `OUTPUT` and `RETURN`.

Candidates need to read each question carefully before attempting to answer it. Answers may address individual topics in a number of different ways.

General comments

There were some excellent programming solutions, but a significant number of candidates demonstrated skill levels suggesting they had limited programming experience.

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

If candidates cross out an answer, they must write any new answers clearly, so that examiners can read the text easily and then award the correct mark.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful if candidates 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 showed a good understanding of the different program constructs. The small number of incorrect answers did not follow any particular pattern.

(b) (i) The majority of candidates answered this question well. The suggestion that variable `MyGrade` should be a String rather than a Character was the most common mistake. Candidates received credit for alternative types to Real for the variable `MyAverage`.

(ii) Many candidates obtained full marks for this question. A common error was to omit the quotation marks for the first answer to differentiate between a string and an identifier name. A number of candidates gave a decimal answer for the second expression and many omitted the hyphen from the string `"Air-con"`.

Question 2

(a) (i) Many candidates had difficulty with the logic and often nested the final conditional clause (testing to see if today was Wednesday) inside the clause testing for `Points > 99`.

Many candidates had not read the question carefully enough and wrote

`IF today = Wednesday` instead of the expression to test for `Today() = 3`.

. Most candidates received the mark for the correct assignment of `0.2` and `0.1` to `DRate`.

A significant number of responses used `OUTPUT` instead of `RETURN`.

(b) (i) The majority of candidates provided the names ‘Syntax’ and ‘Logic’, with some candidates giving textbook descriptions. Many explanations were vague, but in the case of syntax error, an appropriate example often gained a credit.

(ii) Many candidates gave the name ‘Stub testing’, but often, did not provide appropriate descriptions.

(c) (i) Many responses provided variable names instead of values.

(ii) A minority of candidates provided a correct response. Many responses were similar to a normal variable declaration, including a data type, and many omitted the value itself. A significant number of responses simply took a variable name from the identifier table and attempted to re-declare this as a constant.

The majority of candidates achieved one mark for a sensible constant name with a value. A small number of candidates gained both marks.

(iii) Many candidate responses suggest that candidates have limited experience of using constants. The marks awarded were usually for referencing the constants as preventing values being ‘accidentally changed’ or that they were ‘changed in one place only’.

Most answers simply stated that a constant is a value that does not change. This did not answer the question.

(iv) Many of the responses referred to the standard textbook answer of ‘tried and tested’. Many vague responses suggested the use of library routines to “reduce the chance of errors”. Reference to library routines ‘helping to find errors’ was occasionally seen.

(v) Many candidates did not read this question carefully. The context was ‘when writing a program’. This ruled-out activities that would take place either before or after this stage. Many candidates also did not ‘explain how this helps’.

Common incorrect answers were the use of subroutines, stepwise refinement, and modular programming.

Question 3

(a) Around half of the responses correctly referred to the computer being only able to understand machine code. The term ‘understand’ is not sufficient but the reference to machine code gained the mark on this occasion. Many answers were not precise enough to gain credit at this level. Examples include a computer ‘cannot understand high level language’ or ‘can only read binary’. Some candidates confused the terms ‘machine code’ with ‘object code’.

(b) Many candidates stated correct methods for the first mark point but few could correctly describe the method. Dry runs/trace tables were often given for a correct method but the explanations were generally not adequate. Those candidates citing breakpoints, stepping or variable watch windows often gave descriptions that were more precise.

(c) This question was generally well answered, with most candidates gaining the first two marks.

Question 4

(a) (i) The first three rows were often correctly answered, with the first row being correct in the majority of cases. Only a small number of candidates gave the correct answer for the final part.

(ii) Most answers gained one mark for stating that `SearchString` is compared to a string from the array and many correctly identified the loop structure. Only a minority of candidates identified that it was a 2D array and therefore could not describe how the string was extracted. Several answers referred to the `RETURN` (or in some cases `OUTPUT`) but this activity is not included in the range of lines given in the question.

(b) The question presented a relatively simple task of converting pseudocode to program code with minor modification. There were a number of very good responses, but in general, a wide range of marks were awarded.

There was confusion in all languages regarding the use of functions and methods to convert a string to upper case or lower case.

The following syntax errors were very common:

VB:

- Use of ‘`Returns Integer`’ instead of ‘`As Integer`’ in function header.
- Incorrect brackets for array indices.
- Use of `ENDFOR` to terminate the loop.

Python:

- Incorrect array referencing.
- Incorrect number of iterations (e.g. `range(1, NumElements)`).
- Single “=” symbol in comparison statement.
- The use of “`Call`” in the statement `SaveToFile(ArrayString).`

A number of candidates offered solutions that were clearly pseudocode.

(c) Candidates generally answered this question well, with many gaining one mark for the name and another for stating that the modules would be easier to test or debug.

(d) Most responses included a nested `FOR` loop for processing the 2D array. Some candidates implemented a single loop that included separate assignments for the two elements in the given row/column and this was perfectly satisfactory for the given array.

Many candidates used incorrect array syntax, and in several cases, the number of iterations was one less than required.

Question 5

There were only a small number of excellent solutions for this question.

A common mistake was to attempt to pass parameters into the function, which contradicted the first bullet point in the question. In many cases, the concatenation to form the text string to write to the file was incorrect, often missing the comma separators.

The syntax used for the calls to the three functions given in the question was often incorrect, and in many cases, the return value was unassigned.

Some perfect solutions to the loop were seen, but many looped either two or four times or in some cases indefinitely.

A significant number of answers included the `OUTPUT` of a Boolean rather than its `RETURN`.

Few solutions could correctly write pseudocode to open, write, and then close a file.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and be able to apply them to the scenarios presented if they are to achieve high marks.

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 important that candidates writing program code use the correct syntax for their chosen language and they must show a good understanding of fundamental programming concepts. Examples include the difference between a literal and an identifier as well as the difference between `OUTPUT` and `RETURN`.

Candidates need to read each question carefully before attempting to answer it. Answers may address individual topics in a number of different ways.

General comments

There were some excellent programming solutions, but a significant number of candidates demonstrated skill levels suggesting they had limited programming experience.

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

If candidates cross out an answer, they must write any new answers clearly, so that examiners can read the text easily and then award the correct mark.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful if candidates 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 showed a good understanding of the different program constructs. The small number of incorrect answers did not follow any particular pattern.

(b) (i) The majority of candidates answered this question well. The suggestion that variable `Revision` should be a String rather than a Character was the most common mistake. Candidates received credit for alternative types to Real for the variable `MaxValue`.

(ii) Most candidates gained between two and four marks for this question. A common error was to omit the quotation marks for the first answer to differentiate between a string and an identifier name. A number of candidates used the characters from the variable name itself, (`Activity`) rather than the characters it contained (“`Design`”). Some candidates gave a decimal answer for the second expression.

Question 2

(a) (i) Most candidates correctly interpreted the nested `IF` structure.

A common error was to omit the parameters from the function header, and in many cases, the final `RETURN` was nested within the `IF` clause, which meant that the value would not be returned in all cases.

A significant number of responses used `OUTPUT` rather than `RETURN`.

(ii) The majority of candidates received at least at least two marks for this question. Most candidates recognised the need for a loop together with some conditional test.

Common errors included:

- declaring the local variable for the return value as something other than a `REAL`
- omitting the prompt.
- incorrect syntax for the conditional statement
- the use of `OUTPUT` rather than `RETURN`.

An example of an incorrect conditional statement was the omission of the variable name from the second comparison e.g. `IF Value > 0 AND < 10000`. While this may work in some programming languages, it is not generic and not precise enough for pseudocode. A small number of candidates attempted to invert the conditional test (i.e. `IF value <= 0 OR Value >= 10000`) but the conversion did not always work.

(b) (i) The majority of candidates correctly named a Run-time error, although many other general computing terms were suggested. ‘Run-Length’ error was particularly common.

Many candidates found it difficult to provide a description. Many simply referred to the program “crashing or freezing” which as not considered adequate at this level. Most successful responses gave the example of dividing by zero.

(ii) A small number of very good answers were seen, but generally, this was not well answered. Many candidates seemed to have misread the question and gave answers relating to `GetTotal()` rather than `CalcPoints()`.

The descriptions often did not refer to the subsequent activities from the flowchart.

Question 3

(a) Many candidates demonstrated an understanding of the term ‘Transferrable Skill’. A small number of candidates were able to explain **how** the skill would help. Many gained one mark for referring to a specific program feature.

(b) Most candidates achieved one mark for a general description of breaking down a problem or algorithm. Some referred to breaking down an existing program, which was not accepted. A minority of candidates were able to describe well the objective of the process.

(c) There was a wide range of responses for this question. A typical vague response was “the given value is compared through 1000 elements”. Most candidates were able to carry out the comparison of array element with the given value.

The most obvious step missed in most answers was breaking out of the loop if the given value had been found. Responses that simply repeated the bullet points from the question were not sufficient.

Several candidates offered solutions based on either pseudocode or program code.

Question 4

(a) (i) The first three rows were often correctly answered, with the first row being correct in the majority of cases. Answers to the last two rows were often incorrect, with only a small number of candidates gave the correct answer for the final part.

(ii) Most answers gained one mark for stating that `SearchString` is compared to a string from the array and many correctly identified the loop structure. In addition, many answers correctly described the actions following a comparison between matching strings. A minority correctly identified that it was only the first seven characters that were compared, and some thought that the loop would continue until the search string was found.

Several answers referred to the `RETURN` (or in some cases `OUTPUT`), but this activity is not included in the range of lines given in the question.

(b) The question presented a relatively simple task of converting given pseudocode to program code, plus a minor modification. A wide range of marks was awarded.

Many solutions included a great deal of native file-handling operations, which were not required and several solutions omitted the loop.

Common syntax errors included:

VB:

- Use of ‘`Returns Integer`’ instead of ‘`As Integer`’ in function header
- Incorrect brackets for array indices
- Incorrect loop syntax
- Use of `Console.Writeline()` instead of writing values to the array

Python:

- Incorrect use of slice operation to obtain substrings
- Single “=“ symbol in comparison statement.

A number of candidates offered solutions that were clearly pseudocode

(c) (i) Candidates appear to be familiar with the term ‘sub-tasks’, but most could not map these to specific program constructs.

(ii) Many candidates correctly identified a benefit from the list given.

(d) Most answers understood the need for a `FOR` loop to process each element in the array and the assignment of the given value. Many did not use the correct array syntax.

In several cases, the number of iterations was one less than that required.

A number of correct Python solutions did not require a loop.

A number of candidates offered solutions that were clearly pseudocode.

Question 5

A small number of candidates provided excellent solutions

A common error was defining the parameters to the function. In many cases, the file operations were incorrect; relatively few candidates made use of the filename passed in as a parameter.

Most candidates recognised the need for some form of loop. Many of these correctly attempted a pre-condition loop, which tested for end of file or an empty string, but others attempted a simple count-controlled loop designed to output 20 lines at a time regardless of whether there were sufficient lines available from the file.

A small number of candidates attempted to check after every 20 lines. In many cases, the code omitted the necessary `INPUT` statement or did not compare any given input with ‘`N`’ as given in the question. Some candidates were confused between the `StepNumber` variable and the variable used to count the number of lines that had been output.

A small number of candidates gave solutions that simply output the example from the question.

COMPUTER SCIENCE

Paper 9608/23
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and be able to apply them to the scenarios presented if they are to achieve high marks.

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 important that candidates writing program code use the correct syntax for their chosen language and they must show a good understanding of fundamental programming concepts. Examples include the difference between a literal and an identifier as well as the difference between `OUTPUT` and `RETURN`.

Candidates need to read each question carefully before attempting to answer it. Answers may address individual topics in a number of different ways.

General comments

There were some excellent programming solutions, but a significant number of candidates demonstrated skill levels suggesting they had limited programming experience.

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

If candidates cross out an answer, they must write any new answers clearly, so that examiners can read the text easily and then award the correct mark.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful if candidates cross out this text.

Comments on specific questions

Question 1

(a) The majority of candidates showed a good understanding of the different program constructs. The small number of incorrect answers did not follow any particular pattern.

(b) (i) The majority of candidates answered this question well. The most common mistake was the suggestion that the variable `FuelType` should be a string rather than a character. Candidates gained credit for alternative types to real for the variable `MinValue`.

(ii) Candidates are reminded that they must make use of quotes to clarify between a string variable and an identifier. Many candidates correctly identified that the resultant string was "`Month: DEC`" but they omitted the important quotation marks. Some candidates did not apply the `INT` function and therefore gave an answer of 12.6 or gave an incorrect answer of 13 as they assumed that the `INT` function would round up.

Question 2

(a) (i) Most candidates scored full marks. Most were able to write correctly structured `IF THEN ELSE` statements. Many weaker responses used an incorrect function heading and incorrect declaration statements.

(ii) Many candidates provided good pseudocode solutions using either `REPEAT … UNTIL` or `WHILE … ENDWHILE` loops. Some candidates missed out the prompt statement for the user to input a card number.

(b) (i) Run-time or logic error were the responses from many candidates. The descriptions for these, in particular for run-time error were often imprecise. Some responses used examples of situations that cause these errors, which was acceptable for this question.

(ii) Many candidates achieved full marks for this question. Many of those that did not gain full marks had provided values such as characters or real numbers where an integer was expected. The previous question stated that the function did not contain any syntax errors, which means that test data should not have included values such as characters or real numbers for integers.

(c) Those that could correctly identify a type of maintenance could usually give a correct description. It was clear that other candidates had not studied the types of program maintenance.

Question 3

(a) Many of the candidates were able to state a `FOR` loop as being the most appropriate and they were able to justify this. A significant number of candidates appeared to think that 'bubble sort' is the name of a loop structure.

(b) There was a range of different responses from candidates, with many achieving at least one or two marks. Candidates in general either attempted to describe a comparison of the max and min values to each element of the array or tried to describe a bubble sort. Credit was given to either method. Many using the former method initialised the max and min values to random high and low values rather than the first element of the array.

Question 4

(a) (i) Many candidates identified a global identifier and recognised that the array was two dimensional. A common mistake was to give `Update` as a user defined procedure. A minority of candidates showed an understanding of the 'scope'.

(ii) Most answers gained one mark for stating that `NewData` is compared to a string from the array and many correctly identified the loop structure. Few candidates identified that it was a 2D array and could not describe how the string was extracted.

(b) The question presented a relatively straightforward task of amending given pseudocode to program code, with minor modifications. A small number of very good answers were seen, but overall a wide range of marks was awarded.

There was confusion in all languages regarding the logic used for the nested `IF` statement in the amended code.

The following syntax errors were also common.

VB:

- Use of '`Returns Integer`' instead of '`As Integer`' in function header
- Incorrect brackets for array indices

Python:

- Incorrect array referencing

- Single "=" symbol in comparison statement

A number of candidates offered solutions that were clearly pseudocode.

(c) About half the candidates were able to give some description of parameters and the descriptions were precise and detailed for the second mark.

(d) (i) The responses to this question were divided with candidates in general either achieving two or three marks or no marks. Many found it challenging to write code to check whether a character was numeric or not. Python solutions often included the incorrect number of iterations (e.g. `range (1, 200)`). Marks were awarded for some correct Python solutions that did not require a loop.

(ii) Candidates found this question challenging. Many answers were similar to a normal variable declaration using an assignment symbol instead of '=', and `DECLARE` instead of `CONSTANT`

Question 5

Many excellent solutions achieved either the maximum or nearly the maximum mark available for this question. These candidates had clearly read the question and understood what was required.

Three of the marks involved opening, reading, and closing a file, `FileName`. Candidates should be familiar with these common operations. Some candidates used incorrect syntax, such as not including the file name or adding quotes to the file name variable.

Many included a conditional loop but again the syntax seen was often incorrect, such as a missing file name when using the `EOF()` function. Some candidates incorrectly used assignment symbols for a comparison e.g. `IF FileLine ← ""`.

A number of candidates wrote solutions in a combination of pseudocode and program code.

Candidates need to be aware that the key word `OUTPUT` is not the same as `RETURN`.

COMPUTER SCIENCE

Paper 9608/31
Written Paper

Key messages

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

Questions that ask ‘Explain how...’ require candidates to explain how to perform a particular task mentioned in the question rather than what the task is. Questions that ask the candidate to ‘Explain why...’ require a technical explanation of the reasons for a particular result given in the question.

General comments

Candidates need to read each question very carefully before attempting to write an answer. The instructions for **Question 2(d)**, ‘Explain why this is output.’ and for **Question 5(a)**, ‘Explain how the processes are affected...’ require different types of answer.
An example of a good answer is ‘The values 0.1, 0.2 and 0.3 cannot be represented accurately in binary. This in accuracy increases when the three numbers are added together, so the value output is sufficiently inaccurate to be seen.’

The changes in state of the process must be included in the answer to **Question 5(a)** to explain how. An example of a good answer to **Question 5(a)(i)** is ‘The process will be halted then moved to the blocked state.’

Comments on specific questions

Question 1

(a) Most candidates identified the correct data type of `Book`. Others found this part of the question challenging. Better responses correctly identified the data type as a record.

(b) Some candidates correctly identified the data type as enumerated.

(c) (i) Most candidates provided a correct declaration in pseudocode.

(ii) A minority of candidates wrote a correct assignment statement in pseudocode.

Question 2

(a) Most candidates could correctly calculate the denary value and state why the number was not normalised. Normalising the floating-point number, especially providing a correct value for the exponent proved more challenging.

(b) More candidates could correctly complete the mantissa and the exponent for the largest positive number than for the smallest positive number. Some candidates were unaware that a positive number can have a negative exponent.

(c) Most candidates could correctly identify the effect of increasing the mantissa and increasing the exponent.

(d) Most candidates found this part of the question challenging and could not apply knowledge of floating point to providing an explanation of the problem with the calculation given on the question paper.

Question 3

(a) Many candidates found this part of the question challenging and incorrectly stated the tasks performed by a firewall.

(b) (i) A minority of candidates correctly stated three functions of TCP. The most frequent correct response was 'dividing the file into packets'.

(ii) A minority of candidates correctly stated two functions of IP. The most frequent correct response was 'adding a destination address to the IP header'.

(iii) Candidates generally answered this well. FTP was a popular correct answer.

(c) Many candidates identified the three correct protocols.

(d) Better responses applied knowledge of TCP/IP protocol to the sending of an email message. Others incorrectly described other protocols used for sending or receiving emails.

Question 4

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

(ii) Many responses showed a correctly completed Karnaugh Map (K-map).

(iii) Nearly all responses showed correct grouping. The most common error was to incorrectly group the three products in the bottom row.

(iv) Many responses showed a correct simplified sum-of-products for the answer to **part (a)(iii)**.

(b) (i) Most responses showed a correctly completed Karnaugh Map. The most common error was to incorrectly swap the third and fourth labels for the rows and columns.

(ii) Most responses showed correct grouping.

(iii) Providing a correct simplified sum-of-products for the answer to **part (b)(ii)** proved more of a challenge.

Question 5

Some candidates could correctly identify the state of the process after each event. Providing an explanation proved more of a challenge to nearly all candidates.

(a) (i) A minority of candidates stated that the running process needed to be halted before it could move to the blocked state.

(ii) A minority of candidates stated that the running process had to be halted before it could move to the ready state.

(b) (i) A small number of candidates correctly identified that the processor needed to be available and that the process needed to be at the head of the ready queue.

(ii) A small number of candidates correctly identified that all resources needed to be available and all outstanding events completed before the process could move from the blocked to the ready state.

(c) Many candidates showed good knowledge and understanding of the general principle that the use of the processor should be maximised. Few candidates included the detail required for further marks.

Question 6

(a) Most candidates provided some correct entries for the symbol table.

(b) Candidates who had correctly completed the symbol table in **part (a)** usually provided the correct output from the lexical analysis.

(c) A minority of responses were detailed enough to be creditworthy.

(d) (i) A small number of candidates correctly identified that a program with optimised code includes fewer instructions which results in the program completing the task in a shorter time. A common misconception was that the instructions themselves would be executed more quickly.

(ii) Candidates who understood assembly language provided excellent responses to this part of the question. Other candidates found this part of the question challenging. Some responses were completely correct and many were completely incorrect. Some candidates did not attempt this part of the question.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

Key Messages

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

General Comments

Candidates need to read questions very carefully before attempting to write an answer. For example, in **Question 1(a)** and **Question 5(b)(ii)**, candidates needed to show their working to gain the full marks.

Question 2(a) asks for the features of a star topology. Responses that only gave benefits and drawbacks will not gain credit.

Comments on Specific Questions

Question 1

(a)(i) There was a full range of responses to this question with many achieving good marks.

(ii) There was a full range of responses to this question with many achieving good marks.

(iii) Many candidates were able to convert the negative denary number to a normalised floating-point number. Others found this challenging. Most candidates gained some credit for showing a correct conversion to two's complement.

(b)(i) Many candidates correctly identified that this was the largest possible number that could be represented in this format.

(ii) Most candidates identified that overflow would occur.

Question 2

(a) Many candidates correctly identified features of a star topology such as 'each node has a dedicated bidirectional connection to a central device'. Other responses gave advantages of a star topology such as 'fewer collisions', when the question asked for 'features'.

(b)(i) Many candidates correctly identified that a dedicated path was required. Fewer responses then went on to explain that this would need to be set up for the duration of the communication.

(ii) There was a full range of responses to this question with many achieving good marks.

Question 3

(a) Many candidates made some correct use of the laws of Boolean algebra to simplify the expression.

(b) (i) The majority of candidates answered were able to provide a correct response to this question.

(ii) Many responses showed a correctly completed Karnaugh Map (K-Map).

(iii) Nearly all responses showed correct groups of 1s.

(iv) Many responses showed a correct simplified sum-of-products for the answer to **part (b)(iii)**.

(c) (i) Most responses showed a correctly completed Karnaugh Map. The most common error was to swap the third and fourth labels for the rows and columns.

(ii) Most responses showed correct grouping.

(iii) Candidates found it challenging to provide a correct simplified sum-of-products for the answer to **part (b)(ii)**.

Question 4

(a) Most candidates could provide some correct entries for the symbol table.

(b) Candidates who had correctly completed the symbol table in **part (a)** usually provided the correct output from the lexical analysis.

(c) (i) Some candidates correctly identified code optimisation as the final stage of compilation. A common incorrect answer was to quote the last line of the assembly language program on the question paper

(ii) Candidates who understood assembly language provided excellent responses to this part of the question. Other candidates found this part of the question challenging, so some responses were completely correct, and some were completely incorrect. Some candidates did not attempt this part of the question.

(iii) A minority of candidates correctly identified that a program with optimised code includes fewer instructions, which means that the program completes the task in a shorter time. A common misconception was that the instructions themselves would be executed more quickly.

(d) There was a full range of responses to this question. Common errors included placing operators on the stack and an error in the final calculation giving an incorrect answer of −7.

Question 5

Some candidates could correctly identify the state of the process after each event, providing an explanation proved more of a challenge to nearly all candidates.

(a) There was a full range of responses to this question with many achieving full marks.

(b) (i) Most candidates achieved full marks on this question.

(ii) Candidates generally found it challenging to calculate how many clock cycles are saved and showing their working.

(c) Candidates generally answered this question well, with many achieving full marks.

Question 6

(a) Most candidates provided the correct terms. The descriptions proved more challenging as each description needed to be complete and accurate.

(b) Most candidates achieved full marks on this question.

COMPUTER SCIENCE

Paper 9608/33
Written Paper

Key messages

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

Questions that ask ‘Explain how...’ require candidates to explain how to perform a particular task mentioned in the question rather than what the task is. Questions that ask the candidate to ‘Explain why...’ require a technical explanation of the reasons for a particular result given in the question.

General comments

Candidates need to read each question very carefully before attempting to write an answer. The instructions for **Question 2(d)**, ‘Explain why this is output.’ and for **Question 5(a)**, ‘Explain how the processes are affected...’ require different types of answer.
An example of a good answer is ‘The values 0.1, 0.2 and 0.3 cannot be represented accurately in binary. This in accuracy increases when the three numbers are added together, so the value output is sufficiently inaccurate to be seen.’

The changes in state of the process must be included in the answer to **Question 5(a)** to explain how. An example of a good answer to **Question 5(a)(i)** is ‘The process will be halted then moved to the blocked state.’

Comments on specific questions

Question 1

(a) Most candidates identified the correct data type of `Book`. Others found this part of the question challenging. Better responses correctly identified the data type as a record.

(b) Some candidates correctly identified the data type as enumerated.

(c) (i) Most candidates provided a correct declaration in pseudocode.

(ii) A minority of candidates wrote a correct assignment statement in pseudocode.

Question 2

(a) Most candidates could correctly calculate the denary value and state why the number was not normalised. Normalising the floating-point number, especially providing a correct value for the exponent proved more challenging.

(b) More candidates could correctly complete the mantissa and the exponent for the largest positive number than for the smallest positive number. Some candidates were unaware that a positive number can have a negative exponent.

(c) Most candidates could correctly identify the effect of increasing the mantissa and increasing the exponent.

(d) Most candidates found this part of the question challenging and could not apply knowledge of floating point to providing an explanation of the problem with the calculation given on the question paper.

Question 3

(a) Many candidates found this part of the question challenging and incorrectly stated the tasks performed by a firewall.

(b) (i) A minority of candidates correctly stated three functions of TCP. The most frequent correct response was 'dividing the file into packets'.

(ii) A minority of candidates correctly stated two functions of IP. The most frequent correct response was 'adding a destination address to the IP header'.

(iii) Candidates generally answered this well. FTP was a popular correct answer.

(c) Many candidates identified the three correct protocols.

(d) Better responses applied knowledge of TCP/IP protocol to the sending of an email message. Others incorrectly described other protocols used for sending or receiving emails.

Question 4

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

(ii) Many responses showed a correctly completed Karnaugh Map (K-map).

(iii) Nearly all responses showed correct grouping. The most common error was to incorrectly group the three products in the bottom row.

(iv) Many responses showed a correct simplified sum-of-products for the answer to **part (a)(iii)**.

(b) (i) Most responses showed a correctly completed Karnaugh Map. The most common error was to incorrectly swap the third and fourth labels for the rows and columns.

(ii) Most responses showed correct grouping.

(iii) Providing a correct simplified sum-of-products for the answer to **part (b)(ii)** proved more of a challenge.

Question 5

Some candidates could correctly identify the state of the process after each event. Providing an explanation proved more of a challenge to nearly all candidates.

(a) (i) A minority of candidates stated that the running process needed to be halted before it could move to the blocked state.

(ii) A minority of candidates stated that the running process had to be halted before it could move to the ready state.

(b) (i) A small number of candidates correctly identified that the processor needed to be available and that the process needed to be at the head of the ready queue.

(ii) A small number of candidates correctly identified that all resources needed to be available and all outstanding events completed before the process could move from the blocked to the ready state.

(c) Many candidates showed good knowledge and understanding of the general principle that the use of the processor should be maximised. Few candidates included the detail required for further marks.

Question 6

(a) Most candidates provided some correct entries for the symbol table.

(b) Candidates who had correctly completed the symbol table in **part (a)** usually provided the correct output from the lexical analysis.

(c) A minority of responses were detailed enough to be creditworthy.

(d) (i) A small number of candidates correctly identified that a program with optimised code includes fewer instructions which results in the program completing the task in a shorter time. A common misconception was that the instructions themselves would be executed more quickly.

(ii) Candidates who understood assembly language provided excellent responses to this part of the question. Other candidates found this part of the question challenging. Some responses were completely correct and many were completely incorrect. Some candidates did not attempt this part of the question.

COMPUTER SCIENCE

Paper 9608/41
Written Paper

Key messages

Candidates need to demonstrate a range of skills and write both program code and pseudocode in a variety of different paradigms, as well as completing common algorithms for data structures.

General comments

Candidates generally performed well in the declarative language questions. They demonstrated a good understanding of how to create clauses, queries, and rules.

Candidates were able to create procedures following the instructions. Others did not demonstrate how to access and use random access files. For question 2, many candidates attempted to access the data as though it was in a traditional text file and could not make suitable use of the Hash() function to gain the address.

Candidates showed a good understanding of stacks and were usually able to show the result of the processes on the stack and how POP and PUSH algorithms function.

Candidates found the recursion question more challenging. When rewriting the iterative process, many candidates retained the recursive call, or replaced with it with a different recursive call.

Comments on specific questions

Question 1

(a) (i) Many candidates gave two suitable clauses. Some candidates found an alternative acceptable way of writing the clauses. Common errors included incorrect spellings of pre-existing terms e.g. feature, or the introduction of capital letters that were not appropriate.

(ii) Candidates did well on this question and were able to follow through with the terms they used in the previous question part to ensure consistency in the clauses.

(b) (i) Many candidates were able to provide the correct answers to this question. Some candidates gave an additional incorrect result. Some candidates attempted to add additional words (such as AND) that were inappropriate as these would not be returned by the goal.

(ii) Many candidates were able to give feature correctly, but often the variable and tuna were in the incorrect order within the goal. Some candidates added additional incorrect content such as checking whether `C` has gills.

(c) Some candidates found it challenging to complete rule given. Most candidates gave an appropriate AND within their rule, but fewer candidates were able to correctly give both components of the rule. There was a variety of acceptable solutions. Some candidates introduced incorrect capital letters, and/or spellings of terms that would stop the rule from working. Some candidates did not make use of the variable `X`, for example `bird(has_wings) AND bird(lays_eggs)` without reference to that actual animal (`X`) in the rule.

(d) (i) Candidates found it difficult to explain the term programming paradigm. Many candidates instead defined it as the language, or the way that someone programs which explains the effects of the chosen paradigm. The most common appropriate answer was that it was the style of programming.

(ii) Many candidates were able to give at least one paradigm. The most common responses were low-level and imperative/procedural. Some candidates gave a repeat answer, for example, low-level and then assembly language, or imperative and then procedural.

Question 2

(a) Many candidates were able to gain at least two marks for declaring `Fiction` and `LastRead` appropriately. Fewer candidates were able to give a suitable declaration for the ADT `Book`, with some candidates attempting to define a procedure or function. The most common error was in the declaration of ISBN, where many candidates put Integer. The table clearly has the ISBN in speech marks to indicate a string. Candidates should take note of the leading zeros in the example ISBN number and excluding Integer as an appropriate data type. Some candidates attempted to assign data in the declaration, which was incorrect.

(b) Most candidates were able to declare the function appropriately for their chosen language. A significant number of candidates attempted to read the ISBN into the function from input, instead of as a parameter as stated in the question. Some candidates were unable to use modulus division in their chosen language, and some candidates used the variable ISBN and the value 2 000 in the incorrect order. Some candidates returned the original (unchanged) value instead of returning the calculated value. Others output this value instead of returning it.
Few candidates recognised that the parameter would be a string (from the previous question) and would need conversion prior to the calculation.

(c) This question required an understanding of the random files and the previous two parts that set up the structures and functions candidates needed to use. Candidates did not provide a prompt (as required by the question). They read the value in without an output a message for this. Candidates must make sure that they read each question carefully to make sure they meet all of the criteria. Some candidates attempted the validation of the input, most commonly by checking the number of digits. Few candidates checked that the ISBN was completely numeric as well. A common error included checking the ISBN (as an integer) was between 1 000 000 000 000 and 9 999 999 999 999, but this excluded the possible value of 0 000 000 000 000.

Many candidates were able to call the Hash() function, but fewer recognised the need to store the return value.

Opening the data file required candidates to recognise that this was a random access file. Some candidates achieved this but then did not close the file later in the program.

When reading the data from a random access file, candidates needed to show an understanding that the record is accessed in its entirety and stored in an appropriate format. Some candidates attempted to read in multiple pieces of data and store them in separate locations. Where the data is being read into an ADT, candidates need to show that the variable belongs to that ADT i.e. by declaring it. A significant number of candidates attempted to read the data into a variable named `Book` of type `Book`, the variable and data type cannot both have the same identifier.

Question 3

(a) Candidates demonstrated a good understanding of a stack. The most common response was ‘it is a last in first out structure’.

(b) (i) Many candidates were able to accurately identify the purpose of `StackPointer` as storing the next free space. Some candidates were not specific enough, for example stating that it was a free space, or a place where data can be put – and not that it is the next available position.

(ii) A significant number of candidates were not able to correctly follow the POP and PUSH instructions. Some candidates had the correct number of elements but the first two values in the stack were incorrect. Some candidates used a queue structure and removed elements from positions 0 and 1, whilst adding at the bottom.

(c) (i) This question was answered well overall, with many candidates gaining at least some of the marks, commonly for checking if the `StackPointer` was 0. The assignment of a '*' was commonly lost due to not identifying that a '*' is a string and therefore must be within quotes.

(ii) Most candidates were able to gain some marks. The value to compare `StackPointer` was most commonly correct. Some candidates did not use the correct array name from the introduction to the question.

Question 4

(a) (i) This question was answered well with many candidates clearly expressing the meaning of a recursive algorithm.

(ii) Candidates who understood the meaning of a recursive algorithm were able to identify the recursive call line.

(b) There were mixed responses to this question. Candidates were often able to follow the recursive calls, but were then unable to unwind the calls to perform the calculations, often multiplying by `Number` when it was `0`, hence making the calculation incorrect.

(c) (i) The common answer given was that a recursive algorithm may not end if the base condition is never met. Fewer candidates explained the problem associated with this, such as the increase in memory usage required by recursive algorithms could result in stack overflow.

(ii) When rewriting a recursive function as an iterative algorithm, candidates needed to identify the stopping condition and the values that change each time through the loop. Some candidates reinserted the same recursive call within a loop, which meant that they had not removed the recursion and therefore not answered the question. Some candidates achieved the loop and stopping condition but did not have an appropriate starting value, for example, the number being multiplied within the loop each time was 0, and therefore the multiplications resulted in 0.

Question 5

(a) Some candidates found it challenging to express why privates needed to be private. They quite often defined what this meant as opposed to explaining why it was needed. A common misconception was that they were private to stop access from outside of the class, but other classes and the main program can still access the properties, but only through the methods – this distinction demonstrating the understanding that the private setting restricted direct access to give control to how they could be used.

(b) Many candidates were aware of their chosen language's constructor method and were able to use this correctly. Candidates need experience of writing classes in their chosen language, including the passing of data to the constructor to instantiate an object.

Some candidates attempted to read in the number and shape, instead of passing them as parameters. A significant number of candidates could not write a suitable conditional statement to compare the value of number and shape. Common incorrect answers included if number < 0 or > 9, and if shape <> "square" or "triangle" or "circle", both missing the variable in each of the comparisons. Candidates should have extensive experience of writing selection statements in their chosen language prior to the examination.

(c) There were mixed responses to this question. It was clear that some candidates were aware of the purpose and function of a get method. A common error was writing the functionality of a set method, i.e. taking a value and writing it to the property, instead of returning the value. Some candidates accessed the data but output it instead of returning it to the calling program.

(d) Many candidates were able to instantiate an object in their chosen language. Some candidates were able to use the correct code but then did not send the required number and shape to set up the card, with some candidates attempting to set these up manually by directly accessing the (private) properties.

(e) Most candidates were able to write a suitable function header, taking the cards as parameters as required by the introduction. Some candidates were able to compare the cards, by either comparing the cards or the number and then the shape. Candidates attempted to access the data about the cards in a variety of ways, but few used the get methods as required due to the properties being private. Some candidates struggled to compare the numbers of the cards, often comparing the cards as objects as opposed to only accessing the number property. A significant number of candidates were able to work out which number to return. Others did not always access this value correctly. The selection of this value was usually appropriate.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

Candidates need to demonstrate a range of skills, including the interpretation and writing of declarative language code, and object-oriented program code. They also need to apply their knowledge of these paradigms to the given scenarios. Candidates need to be able to define and use user-defined data types. They must then be able to manipulate these in programs.

General comments

Most candidates showed a good understanding of how to read and write the facts and clauses in declarative language questions. Many found the writing of rules more challenging.

When writing object-oriented program code, candidates need to recall their program specific methods and use these within the given class descriptions.

Candidates had a good understanding of PERT and GANTT charts and were able to apply their knowledge appropriately.

Candidates found it challenging to complete and then improve the efficiency of the algorithm for a bubble sort. Candidates need to write and manipulate standard algorithms in order to learn how they work and identify the problems encountered during their writing.

Comments on specific questions

Question 1

(a) A significant number of candidates were able to identify the key components required in the JSP structure diagram. Many candidates were able to split the program appropriately, with components in suitable places. The stronger responses were able to use the selection and iteration symbols correctly and appropriately, only placing these on the elements that could be repeated or that were optional.

(b) Candidates found this diagram more straightforward than the previous one in **part (a)**, with most candidates submitting the correct boxes leading from appropriate sections. Some candidates did not include sufficient selection, for example, they only placed these on the final boxes, not identifying credit and debit as also being optional elements.

Question 2

(a) Most candidates correctly identified the two statements. Some candidates made some minor errors, such as not using the right case, or introducing a space between standard and poodle. Candidates need to demonstrate an understanding of the syntax required for declarative languages and understand that a space is a character.

(b) Most candidates answered this part well. Some candidates did not provide the results as written in the program. They introduced different characters or missed the underline character ('_'). Candidates should be aware that the program would output the exact data it stores. They need to demonstrate this in their answers.

(c) Some candidates found this question challenging, particularly on the data to include within the brackets. Most candidates were able to get the correct function, but often the values were the wrong way around. Candidates need to read the table in the question to identify the purpose of both arguments, and how these work in relation to each other.

(d) Most candidates were able to identify that the logical operator AND was required. Some candidates correctly gave both parts of this rule. A common error was putting the variables the wrong way around such as `is_a(Z,X)`. The question states that `X` is a dog that within the `is_a` clause is the first element. Some candidates followed the rule given in the example code and copied the `type(Z,Y)`, even though it was not relevant to this rule.

(e) Many candidates were able to provide a correct response to this question part. A common error was attempting to identify all of the dogs that were not Labradors or retrievers. These candidates appeared to be interpreting the English sentence instead of the dry running the program code.

Question 3

(a) There were mixed responses to this question. The most common correct answers were the completion of the last three spaces for the swapping of the data values. Fewer candidates were able to identify the correct end conditions for the loops. Some candidates introduced unknown variables such as `Max`, or `LastElement` without any reference to the array or where these values came from. Where candidates used the `List` appropriately, the values were commonly 1 or 2 values out which would stop the loops from functioning. The stronger candidates had often run the algorithm using test data alongside the code to see what it did and to test different solutions to determine the working conditions. When completing the condition statement, a common error was attempting to use the loop counters `Inner` and `Outer` without any reference to the actual array, for example, `IF Inner > Outer THEN`.

(b) (i) Candidates who gave a correct answer to the selection statement in **part (a)** often provided a correct response here too. Candidates needed to consider what they had written and how this would affect the algorithm. Candidates who had not used the array appropriately in **part (a)** could not determine the order because the algorithm was not functional.

(ii) Many candidates were able to identify the line that needed to be changed gave the correct change. Many candidates gave an incorrect IF statement, but then followed through with the correct change (usually by swapping the position of the two sides of the operator).

(c) Some candidates found this question challenging. They copied the previous code and attempted to compare all of the elements after the loops to see if they were all in order, which would be the case, because the loops had finished. The most common mark was for showing that a flag was required to identify if there had been any swaps (or not). This was not always updated appropriately; commonly it was changed to True when there was a swap, but then changed to False if the inner loop ran without a swap. This meant that the True for a swap in the outer loop was overwritten which meant that the flag would no longer work. Some candidates used an assignment (←) in the loop condition which was not appropriate, for example, `WHILE NoSwaps ← False`. Some candidates did not reset the flag appropriately, e.g. each time the outer loop iterated `NoSwaps` needed to return to False to check the array for swaps again.

Question 4

(a) Many candidates were able to identify the correct attributes required. On occasions, these had inappropriate data types, such as Boolean for identifying the musical instrument used. When completing the methods, candidates often put in additional methods that were not required, instead of the method that would produce the output. Fewer candidates were able to show the correct inheritance in the diagram. Some candidates did not identify any inheritance and others attempted to use symbols for aggregation instead of an arrow for inheritance.

(b) Most candidates were able to have a reasonable attempt at declaring the `Performer` class using their chosen language. A significant number of candidates were able to identify the attributes as being private and showed them declared in this manner. Some candidates were unaware of the name of the constructor in their chosen language. It is important that candidates have experience of writing classes in their chosen language, and that they are familiar with the correct syntax. Some candidates were able to give the appropriate number of parameters in the constructor and then used these within the constructor to initialise the values. Candidates should be aware of this as the best practice method for initialising an object. Some candidates read these into the program within the constructor and then set these values, which was acceptable in this scenario.

Most candidates were able to provide the correct procedure header, with some candidates identifying that a parameter would be appropriate. A common error was declaring the methods as functions and attempting to return the values instead of overwriting them.

(c) A significant number of candidates were able to identify that the `Acrobat` class inherited from the `Performer` class. Fewer candidates were able to identify the need to override the `Performer` constructor, and send this the appropriate parameters required for it to function. Fewer candidates were able to write the method `PerformerInfo()` despite this being in the class diagram. Some candidates attempted to read in the value for `UseFire` at this point to determine the appropriate message to output – overriding the stored attribute. Most candidates were able to correct output or return the value depending on the method they declared, and could concatenate some of the appropriate data. A common error was not taking the value of `UseFire` into account and only allowing the output/return of one standard message. Some candidates need to be aware of the correct use of speech marks to identify strings and to surround the variable names in speech marks.

(d) (i) A minority of candidates were able to successfully create an instance of an object in their chosen language. Most candidates were able to assign data correctly to the variable identified. There was variety of ways candidates could set the data, for example sending it as parameters in the constructor. Quite often, candidates missed an item e.g. the TRUE for fire, or they did not correctly identify the data as being a string and enclosing it in speech marks.

(ii) This question required candidates to apply their knowledge of inheritance to the circus program. Some candidates gave generic descriptions without any application to the program. Many candidates were able to identify the subclass and parent class correctly and gave a suitable application i.e. the ability to use the same methods.

Question 5

(a) Many candidates were able to correctly complete the GANTT chart. Some candidates did not follow the dependencies, particularly where one task was dependent on two predecessors.

(b) This required a description of how the work could be allocated between the three teams. Candidates needed to identify an appropriate way of assigning work i.e. teams working on tasks that could be done concurrently. Some candidates were able to identify tasks that could be done concurrently, but were unable to apply their knowledge for an additional mark i.e. what happens when there are not any tasks that can be done concurrently.

(c) (i) Many candidates were able to identify the critical path correctly. Some candidates did not follow the instructions in the question to identify the activities. They instead gave the numbers from the PERT chart that did not relate to the activities.

(ii) This question asked about the importance of the critical path, not what the critical path was. Many candidates defined the critical path and how this is calculated, instead of explaining why it is important. Those candidates that did well on this question often thought of the drawback i.e. when one of the critical path tasks is late, then the length of the project increases. A common error was stating that the critical path allows you to calculate the maximum time for the project.

Question 6

(a) (i) Most candidates made a reasonable attempt at declaring a record data type, with many candidates gaining marks for correctly declaring the `Country` and `Pointer` within the record. Fewer candidates were able to give a suitable declaration for the record type. There were many different ways candidates could have done this. Some candidates attempted to declare it incorrectly as a procedure. A common error included declaring the record type and attempting to assign the data from the table to the variables within the record; not every record will have the same data, this is only an example and candidates should be aware that when a record type is declared it is defining the structure of it, and not assigning data.

(ii) There were mixed responses to this question. Some candidates were able to declare an array structure. A large number of candidates put the data type, `ListElement`, as the array identifier, and the identifier, `CountryList`, as the data type. The question required 15 array elements, starting at index 1. Some candidates started at position 0 and went to 16, and some candidates only allowed 14 elements.

(b) Many candidates made a good attempt to answer this question. The identifier, `ThisPointer` was often placed in the correct positions. Fewer candidates were able to complete the remaining spaces. A common error was using the incorrect pointer value to access the array.

COMPUTER SCIENCE

Paper 9608/43
Written Paper

Key messages

Candidates need to demonstrate a range of skills and write both program code and pseudocode in a variety of different paradigms, as well as completing common algorithms for data structures.

General comments

Candidates generally performed well in the declarative language questions. They demonstrated a good understanding of how to create clauses, queries, and rules.

Candidates were able to create procedures following the instructions. Others did not demonstrate how to access and use random access files. For question 2, many candidates attempted to access the data as though it was in a traditional text file and could not make suitable use of the Hash() function to gain the address.

Candidates showed a good understanding of stacks and were usually able to show the result of the processes on the stack and how POP and PUSH algorithms function.

Candidates found the recursion question more challenging. When rewriting the iterative process, many candidates retained the recursive call, or replaced with it with a different recursive call.

Comments on specific questions

Question 1

(a) (i) Many candidates gave two suitable clauses. Some candidates found an alternative acceptable way of writing the clauses. Common errors included incorrect spellings of pre-existing terms e.g. feature, or the introduction of capital letters that were not appropriate.

(ii) Candidates did well on this question and were able to follow through with the terms they used in the previous question part to ensure consistency in the clauses.

(b) (i) Many candidates were able to provide the correct answers to this question. Some candidates gave an additional incorrect result. Some candidates attempted to add additional words (such as AND) that were inappropriate as these would not be returned by the goal.

(ii) Many candidates were able to give feature correctly, but often the variable and tuna were in the incorrect order within the goal. Some candidates added additional incorrect content such as checking whether `C` has gills.

(c) Some candidates found it challenging to complete rule given. Most candidates gave an appropriate AND within their rule, but fewer candidates were able to correctly give both components of the rule. There was a variety of acceptable solutions. Some candidates introduced incorrect capital letters, and/or spellings of terms that would stop the rule from working. Some candidates did not make use of the variable `X`, for example `bird(has_wings) AND bird(lays_eggs)` without reference to that actual animal (`X`) in the rule.

(d) (i) Candidates found it difficult to explain the term programming paradigm. Many candidates instead defined it as the language, or the way that someone programs which explains the effects of the chosen paradigm. The most common appropriate answer was that it was the style of programming.

(ii) Many candidates were able to give at least one paradigm. The most common responses were low-level and imperative/procedural. Some candidates gave a repeat answer, for example, low-level and then assembly language, or imperative and then procedural.

Question 2

(a) Many candidates were able to gain at least two marks for declaring `Fiction` and `LastRead` appropriately. Fewer candidates were able to give a suitable declaration for the ADT `Book`, with some candidates attempting to define a procedure or function. The most common error was in the declaration of ISBN, where many candidates put Integer. The table clearly has the ISBN in speech marks to indicate a string. Candidates should take note of the leading zeros in the example ISBN number and excluding Integer as an appropriate data type. Some candidates attempted to assign data in the declaration, which was incorrect.

(b) Most candidates were able to declare the function appropriately for their chosen language. A significant number of candidates attempted to read the ISBN into the function from input, instead of as a parameter as stated in the question. Some candidates were unable to use modulus division in their chosen language, and some candidates used the variable ISBN and the value 2 000 in the incorrect order. Some candidates returned the original (unchanged) value instead of returning the calculated value. Others output this value instead of returning it.
Few candidates recognised that the parameter would be a string (from the previous question) and would need conversion prior to the calculation.

(c) This question required an understanding of the random files and the previous two parts that set up the structures and functions candidates needed to use. Candidates did not provide a prompt (as required by the question). They read the value in without an output a message for this. Candidates must make sure that they read each question carefully to make sure they meet all of the criteria. Some candidates attempted the validation of the input, most commonly by checking the number of digits. Few candidates checked that the ISBN was completely numeric as well. A common error included checking the ISBN (as an integer) was between 1 000 000 000 000 and 9 999 999 999 999, but this excluded the possible value of 0 000 000 000 000.

Many candidates were able to call the Hash() function, but fewer recognised the need to store the return value.

Opening the data file required candidates to recognise that this was a random access file. Some candidates achieved this but then did not close the file later in the program.

When reading the data from a random access file, candidates needed to show an understanding that the record is accessed in its entirety and stored in an appropriate format. Some candidates attempted to read in multiple pieces of data and store them in separate locations. Where the data is being read into an ADT, candidates need to show that the variable belongs to that ADT i.e. by declaring it. A significant number of candidates attempted to read the data into a variable named `Book` of type `Book`, the variable and data type cannot both have the same identifier.

Question 3

(a) Candidates demonstrated a good understanding of a stack. The most common response was ‘it is a last in first out structure’.

(b) (i) Many candidates were able to accurately identify the purpose of `StackPointer` as storing the next free space. Some candidates were not specific enough, for example stating that it was a free space, or a place where data can be put – and not that it is the next available position.

(ii) A significant number of candidates were not able to correctly follow the POP and PUSH instructions. Some candidates had the correct number of elements but the first two values in the stack were incorrect. Some candidates used a queue structure and removed elements from positions 0 and 1, whilst adding at the bottom.

(c) (i) This question was answered well overall, with many candidates gaining at least some of the marks, commonly for checking if the `StackPointer` was 0. The assignment of a '*' was commonly lost due to not identifying that a '*' is a string and therefore must be within quotes.

(ii) Most candidates were able to gain some marks. The value to compare `StackPointer` was most commonly correct. Some candidates did not use the correct array name from the introduction to the question.

Question 4

(a) (i) This question was answered well with many candidates clearly expressing the meaning of a recursive algorithm.

(ii) Candidates who understood the meaning of a recursive algorithm were able to identify the recursive call line.

(b) There were mixed responses to this question. Candidates were often able to follow the recursive calls, but were then unable to unwind the calls to perform the calculations, often multiplying by `Number` when it was `0`, hence making the calculation incorrect.

(c) (i) The common answer given was that a recursive algorithm may not end if the base condition is never met. Fewer candidates explained the problem associated with this, such as the increase in memory usage required by recursive algorithms could result in stack overflow.

(ii) When rewriting a recursive function as an iterative algorithm, candidates needed to identify the stopping condition and the values that change each time through the loop. Some candidates reinserted the same recursive call within a loop, which meant that they had not removed the recursion and therefore not answered the question. Some candidates achieved the loop and stopping condition but did not have an appropriate starting value, for example, the number being multiplied within the loop each time was 0, and therefore the multiplications resulted in 0.

Question 5

(a) Some candidates found it challenging to express why privates needed to be private. They quite often defined what this meant as opposed to explaining why it was needed. A common misconception was that they were private to stop access from outside of the class, but other classes and the main program can still access the properties, but only through the methods – this distinction demonstrating the understanding that the private setting restricted direct access to give control to how they could be used.

(b) Many candidates were aware of their chosen language's constructor method and were able to use this correctly. Candidates need experience of writing classes in their chosen language, including the passing of data to the constructor to instantiate an object.

Some candidates attempted to read in the number and shape, instead of passing them as parameters. A significant number of candidates could not write a suitable conditional statement to compare the value of number and shape. Common incorrect answers included if number < 0 or > 9, and if shape <> "square" or "triangle" or "circle", both missing the variable in each of the comparisons. Candidates should have extensive experience of writing selection statements in their chosen language prior to the examination.

(c) There were mixed responses to this question. It was clear that some candidates were aware of the purpose and function of a get method. A common error was writing the functionality of a set method, i.e. taking a value and writing it to the property, instead of returning the value. Some candidates accessed the data but output it instead of returning it to the calling program.

(d) Many candidates were able to instantiate an object in their chosen language. Some candidates were able to use the correct code but then did not send the required number and shape to set up the card, with some candidates attempting to set these up manually by directly accessing the (private) properties.

(e) Most candidates were able to write a suitable function header, taking the cards as parameters as required by the introduction. Some candidates were able to compare the cards, by either comparing the cards or the number and then the shape. Candidates attempted to access the data about the cards in a variety of ways, but few used the get methods as required due to the properties being private. Some candidates struggled to compare the numbers of the cards, often comparing the cards as objects as opposed to only accessing the number property. A significant number of candidates were able to work out which number to return. Others did not always access this value correctly. The selection of this value was usually appropriate.