COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

It is essential that candidates indicate clearly to the examiner when an answer continues either on an additional sheet, or on a blank space within the question paper.

This is a technical subject and candidates should use the technical language associated with the subject in an appropriate way. There is considerable misuse of terminology especially between words such as *record* and *file*, or *memory* and *storage*.

Candidates must read each question carefully and use the context provided when answering the questions. A question that begins with 'Describe' requires a different type of answer from one that begins with 'Explain'.

General comments

The questions about logic expressions and the database question were well done. Many candidates found the questions about video encoding and compression more challenging.

Comments on specific questions

Question 1

(a) The majority of candidates were able to match the type of software to the correct description. A small number of candidates need to improve their understanding of the difference between a library program and a utility program.

(b) Many candidates were able to state that this utility software checks the disk for errors and inconsistencies and marks bad sectors as unusable. There was some confusion with checking for errors and checking for viruses. Some candidates need to understand that restoring data from a backup is a different process from repairing the data on a damaged disk. Responses that simply re-iterate the question such as ‘disk repair software repairs a disk’ are far too vague.

Question 2

(a) There were many correct answers to this question. Some candidates correctly used a single NOR gate to represent NOT (A OR B). Some candidates need to improve their understanding of the XOR gate.

(b) The majority of candidates answered this question well. As explained above, some candidates need to improve their understanding of operation of the XOR gate.

Question 3

(a) A minority of candidates provided a completely correct answer to this question. A number of candidates recognised value1 as a JavaScript identifier, but frequently candidates took the second identifier from the lines of HTML code rather than the JavaScript code. Some candidates need to improve their understanding of the difference between HTML and JavaScript. Some candidates also need to understand that JavaScript identifiers are case sensitive, so ‘`Multiply`’ is a different identifier from ‘`multiply`’.

(b) A minority of candidates provided correct answers to this part question. The answer seen most often was 81 that is 9 * 9 where candidates had overlooked the incrementation of the variable `value1` on line 14 before the multiplication.

(c) Candidates need to improve their understanding of simple JavaScript code. A minority of candidates understood that the statement `value1++` incremented the value of the variable `value1`.

(d) Some candidates correctly recognised that the variable would no longer be converted to an integer. A minority of candidates also understood that adding two string values would result in concatenation rather than arithmetic addition. An example answer is 'The variable `value1` will not be converted to an integer, it will be a string data type, and therefore instead of adding the two values they will be concatenated. For example if value1 is 6, the output would be `66`.'

Question 4

(a) (i) Many candidates need to improve their understanding of the features of a relational database. An example of a good answer is 'In a relational database, multiple tables are linked together which reduces data redundancy and improves data integrity. Different users can be given different views of the data, so they do not see confidential information and data privacy is maintained'.

(ii) The majority of candidates were able to complete the given statements.

(b) (i) The majority of candidates were able to complete the E-R diagram.

(ii) A minority of candidates answered this question well. Many candidates need to improve their understanding of the SQL CREATE DATABASE command.

(iii) A minority of candidates answered this question well. Many candidates also need to improve their understanding of the SQL CREATE TABLE command.

(iv) There were more correct answers to this part question. Some candidates need to improve their understanding of Data Manipulation Language (DML) statements in SQL, particularly the order in which the statements are used.

Question 5

(a) Many candidates were able to state that encryption would be one way of preventing the installation of illegal copies of the software. Fewer candidates were able to correctly state a second method. Some candidates need to understand that copyrighting software does not actually prevent the installation of illegal copies of software.

(b) This question was generally answered well by most candidates. Many candidates stated that the way to distribute the software would be by issuing a compiled version, or executable file.

(c) (i) Some candidates should understand that using a commercial licence does not prevent further copies being made.

(ii) The majority of candidates were able to name two other correct types of software licence.

Question 6

(a) (i) Most candidates were able to identify whether the devices were used for input, for output or for both.

(ii) The majority of candidates could identify the correct sequence of events.

(b) (i) Many candidates provided vague answers to this question. A typical answer was 'to store data'. Candidates need to understand that data is also stored in primary memory, so an answer like this is not precise enough at this level of study.

(ii) A significant number of candidates were able to describe solid-state memory as non-volatile storage with no mechanical or moving parts. Candidates need to improve their understanding of the use of solid-state memory as internal secondary storage.

(c) The majority of candidates were able to state the correct purpose of both RAM and ROM.

(d) (i) Many candidates found this question challenging. Candidates should understand that when describing how an image or a sound is encoded, it is not enough to say 'use an analogue to digital converter'. An example answer is 'The images are stored in bitmap format and are made up of a number of pixels. Each pixel is of one single colour and each colour is encoded by a unique binary number'.

(ii) Many candidates found this question challenging. There was some confusion between the two terms. Candidates need to improve their understanding of interlaced and progressive encoding of videos.

(e) (i) The question asked **how** bit-streaming is used and many candidates need to understand the way to answer this question. Frequent incorrect answers described live or on-demand bit streaming instead of describing how bit streaming works. An example answer is 'Bit streaming is the transmission of the video as a sequence of bits. The video would be compressed and uploaded to a media or web server. On download the server sends the data to a buffer on the receiving computer'.

(ii) The majority of candidates were able to identify that the video would be streamed on-demand, because it was pre-recorded and because the colleagues could then watch at a time convenient to themselves.

(iii) Many candidates found it challenging. Some candidates need to be aware of the requirement for precision when describing the terms. It is not enough to say, for example, that 'temporal redundancy is when pixels have the same value in two different frames'. It must be clear that the pixels that have the same value are in the same location in consecutive frames. An example answer is 'temporal redundancy is when pixels in the same location have the same value in two or more consecutive frames'.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to apply these skills to the scenarios presented. This is a technical subject and makes use of many technical words and phrases. It is important that candidates these correctly.

It is important that candidates writing program code use the correct syntax for their chosen programming language. The understanding of fundamental programming concepts is essential. Examples include the difference between a literal and an identifier and the difference between `OUTPUT` and `RETURN`.

Candidates need to read each question carefully before attempting to answer it. Questions may address topics in various ways and it is often necessary for candidates to respond in context.

General comments

Many candidates performed well on this paper, indicating a good, sound understanding of the subject. There were many excellent solutions seen to questions that required pseudocode and program code. Candidates generally made good use of built in functions and methods of the programming language. It is important that candidates, who use Python, take care to maintain the correct indentation.

Candidates need to read questions carefully. It was clear from a number of responses that many candidates had not read questions carefully enough. If candidates cross out any answers, they must write any new answers clearly. Many candidates make use of blank pages for rough work when preparing their final response. In these cases, it is extremely helpful if candidates cross out any rough work.

Comments on specific questions

Question 1

(a) (i) Many candidates successfully answered this question. Many candidates provided clear and precise answers. Others used imprecise statements such as ‘good variable names’ or ‘spaces’.

Other errors included candidates referring to specific programming features such as ‘declaration’ or ‘a loop’ and a small number of answers simply repeated lines from the original pseudocode.

(ii) The majority of candidates gained at least two marks: one for `START` and `END` plus one for the first two assignments.

Many candidates made a reasonable attempt at the loop and the test for `Tries > 3`.

The loop part of the flowchart appeared to be the most problematic. Often the conditional check was only partially implemented. Many candidates attempted to combine the initial check (corresponding to the pseudocode statement `IF NOT Full`) with the check that was required at the start of each loop.

Many candidates did not use the flow arrow lines and many used decision diamond boxes that had only one output. Some candidates missed the `OUTPUT` command word.

(b) (i) The majority of candidates answered this question part successfully.

(ii) There was a variety of responses to this question. A common error was to omit the quotation marks from the expressions that evaluated to an object of type `STRING`. Many candidates did not convert the final value to an integer

Question 2

(a) Many responses stated that source code would be created using an editor or that source code was 'understandable by humans', but very few related the source code to an activity in the program development cycle.

Many candidates confused source code with object code.

Many successful responses stated for corrective maintenance as 'debugging' or by making reference to 'finding and fixing errors'.

(b) Most candidates correctly referred to the ability of an IDE to highlight a syntax error. Many responses were imprecise, for example by stating 'indentation' instead of 'auto indentation'.

A significant number of candidates seemed to have missed the fact that the question was asking about initial error detection and gave debugging features such as breakpoints, stepping and watch windows.

Question 3

(a) A minority of candidates provided a correct response to this question part.

Many candidates were unable to give a sensible array declaration. A common error was to give the length of `InString` for the upper array dimension.

Many candidates recognised that a `FOR ... ENDFOR` loop was involved in initialising all array elements. Often the array assignment within the loop was incorrect; in several cases the character '0' being used rather than the value zero.

(b) The majority of correct response included an attempt at the first `FOR ... ENDFOR` loop and the selection of each character in turn from `InString`. Some candidates used `LEFT()`instead of `MID()` to perform the selection.

Only a minority of candidates were able to identify the array element to increment. Many solutions simply incremented a single count variable.

Candidates were expected to use a separate second loop to perform the output. Solutions based on this simpler structure with two independent loops were more successful than those that attempted to use a single, nested loop.

Question 4

(a) There were many good answers to **part (b)** of this question, suggesting that the scenario had been understood.

Only a minority of solutions included the parameter value in the procedure heading. Some solutions confused the use of `MaxVol` and `VolLevel`. A common error was to include a call to the procedure being written.

Several solutions included parameter validation, although the final sentence of the question stated that this was not necessary.

The better solutions tested the value of the parameter and attempted to adjust `VolLevel` accordingly. Ensuring that `VolLevel` remained in the given range was only included in a few solutions, and even here it was usually limited to checking that value was still >= 0 when turning the volume down. Very few solutions correctly addressed how the upper volume limit could be determined.

Many solutions included output statements, which were not required and which, arguably, would have no place in the given scenario.

(b) Most candidates provided a correct response to this question part.

(c) (i) Most response correctly identified either a logical or run-time error and of these many also included a correct description.

(ii) A minority of candidates provided a correct response to this question part. The concept of stub testing does not seem to be widely understood.

Many answers described testing of a modular program in general, commonly referring to the idea of testing each part of the program separately. References to the testing of an incomplete program were rare. Some responses occasionally mentioned use of a dummy module.

Question 5

(a) This question attracted heavily polarised responses. Many candidates provided perfect solutions. A small number of responses bore no relation to a structure chart.

A relatively common mistake was not to represent the correct sequence. Some confusion was evident as to whether circles on arrows should be filled or not, with some candidates drawing circles which were either too small to identify or appeared to be partly filled.

A minority of candidates used the correct double-headed arrow for the variable passed by reference.

Question 6

(a) A minority of candidates provided a correct response to this question part

Most responses included an attempt at a function heading and some form of loop.

The task was simply to write program code to perform a linear search of a 1D array. Many solutions attempted to include unnecessary file handling.

Many candidates chose to `OUTPUT` rather than `RETURN` the Boolean value.

A common mistake was to continue searching through the array even after the value was found. In many cases, this included assigning `FALSE` to the return value after each comparison, which meant that the return value would be incorrect, unless the searched-for value was the last one in the array.

(b) A minority of candidates provided a correct response to this question part.

Most responses included an attempt at a function heading and some form of loop, together with statements for opening and closing files, incrementing a count variable, and returning the count.

The majority of solutions recognised the need to extract the `Reference` from the string read from the file, but often the length calculation was incorrect.

Incorrect syntax was common when calling the `SearchLeavers()` function, with the majority attempting to call it as a procedure. This had the consequence that there was no return value to test.

(c) A minority of candidates provided a correct response to this question part.

Many candidates seemed to have overlooked the final sentence of the question, which stated:

Write a statement in **program code** that uses `CountTimes()` to assign the count of unused elements to the variable `Result`.

The last part of the sentence should have directly led to the left-hand side of the answer statement to give the first mark but this did not happen in the majority of cases. Some candidates used function headers and ad-hoc lines of code.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

Key messages

Candidates need to show an in-depth study of the topics and make good use of appropriate technical terminology on this paper. Candidates, who have studied the theory and have practised the precise use of these tools and techniques, were successful in solving problems on the examination paper.

General comments

Candidates need to read questions very carefully before attempting to write an answer.

For example, in **Question parts 8(a)(i)** and **8(a)(ii)** the answer must be shown in binary, in **Question parts 7(a)(i)** and **7(a)(ii)**, types of data stored in the keyword and symbol tables were required, examples of data were not sufficient for answers to these questions.

Candidates need to show an understanding of user-defined data types in their answers to **Question 6(a)**. Many candidates gave responses that only gave benefits.

Comments on specific questions

Question 1

(a) Most candidates correctly identified some of the terms from the descriptions given. Common errors included incorrectly stating term B was a digital signature.

(b) (i) Many candidates partially explained the purpose of a digital signature. Fewer responses explained that as well as ensuring authenticity, a digital signature ensures that a document has not been altered during transmission.

(ii) Candidates who understood the process of producing a digital signature provided excellent responses to this part of the question. Other candidates found this part of the question challenging. Some candidates did not attempt this part of the question.

Question 2

(a) (i) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks. A common error was to write $\overline{A.B.C}$ for $\bar{A}.\bar{B}.\bar{C}$.

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

(iii) Many responses showed correct grouping. The most common error was the lack of understanding that the two groups overlapped.

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

(b) Candidates who understood the process of simplification using De Morgan's laws provided excellent responses to this part of the question. Other candidates found this part of the question challenging. Some candidates did not attempt this part of the question.

Question 3

(a) Most candidates correctly identified at least two of the four terms required to complete the description of sending a message on a bus network. The protocol required was the term least likely to be correct.

(b) Many candidates were able to provide a correct response for this question regarding the how a router and a Network Interface Card are used in the operation of a bus network.

(c) (i) Many responses were imprecise. A common incorrect answer was router.

(ii) A minority of candidates gave appropriate descriptions that applied to a wireless connection sending and receiving data. Other candidates found this part of the question challenging.

Question 4

(a) A minority of candidates gave good descriptions of virtual memory. The most common error was to describe a virtual machine.

(b) (i) Candidates who understood how paging is used to manage virtual memory gained good marks. Other candidates found this part of the question challenging. Some candidates did not attempt this part of the question.

(ii) A minority of candidates gave a suitable page replacement algorithm. A common incorrect answer was thrashing. Some candidates did not attempt this part of the question.

(iii) Some candidates correctly described the continuous swapping of the same pages from disk to RAM and vice versa. A common incorrect answer was the data held on the disk had been corrupted.

Question 5

(a) The majority of candidates identified a difference between monitoring and control systems. Fewer responses went on to provide a description. The most common error was to describe a virtual machine.

(b) (i) Candidates who explained that it was easier to append each reading to the end of the file, so they are stored in chronological order gained good marks. The most common error was to incorrectly add files instead of readings or records.

(ii) A minority of candidates explained how sequential access could be used for the temperature readings file. A common incorrect answer was to search on the key field.

(iii) Most candidates correctly identified the method as random access. There was a full range of responses was seen for the description with some achieving good marks.

Question 6

(a) Many candidates correctly stated why user-defined data types are necessary. A common error was to state an advantage of using user-defined data types. For example, 'They will make a program less error prone.' is an advantage and gains no credit.

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

(ii) Most candidates wrote a correct declaration statement.

(iii) Many candidates wrote a correct assignment statement.

Question 7

(a) (i) A minority of candidates correctly identified two types of data stored in the keyword table. A full response identified reserved words and operators with their matching tokens.

(ii) A minority of candidates correctly identified two types of data stored in the symbol table. A full response included identifier names, the data type and role. A common error was to include operators.

(iii) A small number of candidates explained the use of the keyword and symbol tables during translation. Other candidates found this part of the question challenging. Some candidates did not attempt this part of the question.

(iv) Many candidates correctly stated an additional task completed at the lexical analysis stage. A popular correct response was the removal of comments. Some candidates did not attempt this part of the question.

(b) Many responses showed good reasons why code is optimised.

Question 8

(a) (i) Most candidates correctly identified the binary value of the exponent. A common error was to state the denary value of the exponent.

(ii) Most candidates correctly identified the binary value of the mantissa. A common error was to state the denary value of the mantissa.

(iii) Most candidates correctly identified that the number stored was positive and gave a correct justification for their answer.

(iv) Some candidates correctly converted the number shown in **part (a)(i)** to denary and showed their working. A common error was to incorrectly use a positive value for the exponent.

(b) Most candidates correctly stated the effects of the change.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

Candidates should have experience of programming a range of abstract data types, and should be able to both program solutions for these ADTs as well as provide descriptions for example of how to add an item to a specific ADT. Candidates should be prepared for these ADTs to be given in a variety of ways, and apply their understanding to the one they are required to manipulate.

Candidates need to continue to practise implementing classes and manipulating objects within a program.

Candidates must be reminded that they will not be awarded any marks if they do not use VB.NET, Pascal/Delphi, or Python programming languages when answering programming questions.

General comments

The majority of candidates were able to access all questions on the paper, and many gave correct responses to the practical activities such as completing the state-transition diagram, and the Program Evaluation Review Technique (PERT) chart.

Candidates appeared to have had experience of defining classes and using objects, and had a better understanding of the purpose of the constructor, get and set methods.

Comments on specific questions

Question 1

(a) (i) The majority of candidates were able to write the correct activities and durations on the lines. Some candidates incorrectly added activities to the dummy activity lines, and some added together the duration to create a cumulative duration.

(ii) Many candidates found this question challenging. A minority of candidates were able to give the accurate purpose of a dummy activity. Common inaccurate answers included an activity that does not have a time, or an activity that cannot be done before another. These are both inaccurate because it is not an activity, it is used to show that the dependencies for the next activity.

(b) Most candidates were able to give the correct actions. Candidates should be familiar with the range of symbols that can be used to identify the actions, e.g. ticks, crosses, dashes etc. Some candidates attempted to write the percentages in the boxes where they were applied, or use +s and –s to identify additions or subtractions, which where they were obvious as to their use were permitted but this should not be an ongoing usage.

(c) (i) This question required candidates to read the descriptions and identify the missing attributes and methods. Many candidates attempted to add an attribute and method for the weekly payment, not identifying from the methods already given that there was a need for `HoursThisWeek`.
Few candidates considered the inheritance between the classes; of those that did, many of the inheritance additions were correct. Some candidates drew the arrows the wrong way, and others attempted to show containment which was not appropriate here.

(ii) Many candidates were familiar with the constructor for their chosen language, and many used this appropriately. The question only required candidates to write the constructor, but many attempted to write the whole class definition, or at least the attribute declarations unnecessarily. Some candidates did not appear to understand the purpose of a constructor, and the actions that it should contain. Candidates should be familiar with writing constructors for a range of classes, including identification of its purpose to initialise attributes within the object.

(iii) Many candidates appeared familiar with the purpose of a get method and were able to accurately define one to return the required value. Some candidates attempted to send a parameter to the function and then returned the same value. Candidates should be able to write get methods for a given class and should be familiar with their purpose.

(iv) Candidates should be familiar with set methods and their purpose. A common error by candidates was to attempt to read in a value from the use within the method. A set method should take a given parameter to allow for encapsulation. Some candidates put the assignment the wrong way around, for example, by assigning the parameter the value of `EmployeeID`.

(v) This question required candidates to validate a parameter and to only set the value when the parameter is valid. Candidates found this question challenging, and often when the value was valid, they returned `True` before setting the value to `Pension`. This meant that the code returned before running this assignment and therefore not meeting all of the requirements.

(vi) There were a mix of responses to this question, which required candidates to use the object and methods they had defined within the main program. Candidates often found the calculations challenging, and attempted to multiply a value by a percentage e.g. 3% as opposed to the actual calculation 3/100 or 0.03.

Some candidates did not allow for all three values to be checked each time. For example, the pension was only checked when they had less than 160 hours, which meant that the calculation was not applied for all appropriate values.

Candidates should be familiar with using methods to access data from objects. A minority of candidates were able to accurate use the get methods to access the data from the given object, with many attempting to pass these values to the function as parameters instead of passing the object from which the values could be accessed.

(d) Most candidates were able to give the correct response.

Question 2

(a) This question required candidates to demonstrate an understanding of a circular queue. Candidates found this challenging and few gave correct values to the algorithm. A minority of candidates were able to correctly identify the constant value for the last index, with common answers including 0 and 8.

(b) Many candidates were able to gain some marks for a standard description of a queue, most commonly for checking if the queue was empty, and accessing the data at the start pointer. Few candidates were able to gain the mark for the circular queue, that if the start pointer exceeds the last index then it returns to the first index. Candidates should be familiar with using the range of ADTs and be able to describe the steps and write algorithms for additions and deletions.

(c) Most candidates were able to identify at least two ADTs. Some candidates repeated a queue, or circular queue, despite the question requiring other ADTs. The most common answers included stacks, trees and linked lists.

Question 3

(a) Many candidates were able to identify at least two different types of test data. Some candidates mixed types of test data with types of test, giving answers such as black box etc.

(b) (i) This question was usually answered well, breakpoints was most commonly given correctly, and there were a range of alternate names for stepping that were given by candidates.

(ii) Some candidates found this question challenging. The most common response was variable watch windows and many candidates gave a suitable description of this. As with **part (a)**, some candidates confused debugging features with types of test, giving responses such as black box testing.

Question 4

This question was answered well by the majority of candidates who were able to complete the diagram accurately. Some candidates attempted to add additional lines such as between Checking PIN and Transaction cancelled, or transitions that looped on a state.

Question 5

(a) Responses to this question were mixed. Many candidates were able to gain the loading, storing and end statements, but fewer could correctly identify a shift to multiply a value. Right shifts were often seen, and values other than #2 such as #1 and #4. Candidates should be familiar with a left and right shift, and the effect these have on a binary number.

(b) Many candidates were able to complete some of these parts accurately. Some candidates attempted to increment and decrement the memory locations e.g. `INC COUNT` instead of loading the value, incrementing the accumulator and then storing the new value. Many candidates did not recognise the need for indexed addressing of `STRING` to access each value from it in turn, instead using `LDD` to access its value.