Qualification Accredited

OCR
Oxford Cambridge and RSA

A LEVEL

Examiners' report

COMPUTER SCIENCE

H446

For first teaching in 2015

H446/02 Summer 2024 series

Contents

Introduction

Our examiners' reports are produced to offer constructive feedback on candidates' performance in the examinations. They provide useful guidance for future candidates.

The reports will include a general commentary on candidates' performance, identify technical aspects examined in the questions and highlight good performance and where performance could be improved. A selection of candidate answers is also provided. The reports will also explain aspects which caused difficulty and why the difficulties arose, whether through a lack of knowledge, poor examination technique, or any other identifiable and explainable reason.

Where overall performance on a question/question part was considered good, with no particular areas to highlight, these questions have not been included in the report.

A full copy of the question paper and the mark scheme can be downloaded from OCR.

Paper 2 series overview

Paper 2 is about algorithms and problem solving. It tests candidates' computational thinking ability to analyse and solve problems. Candidates are expected to be able to write algorithms fluently in either pseudocode or program code and to be able to trace algorithms. This means that candidates require a solid grounding in standard data structures backed by practical experience. The paper also covers Object Oriented Programming (OOP) principles and candidates need to have extensive experience of OOP to be able to tackle section B of the paper successfully.

Candidates who did well on this paper generally:	Candidates who did less well on this paper generally:
• displayed an ability to competently interpret and write pseudocode • demonstrated a clear understanding of OOP principles allied to an ability to write code that instantiated classes and provided methods for a given scenario • displayed a good knowledge of a range of data structures and standard algorithms for operations on the given data structures. The data structures covered were queues, stacks, trees and graphs • where Level of Response questions were set, it was noticeable that more successful candidates tried to make more AO3 evaluative comparisons that were relevant to the scenarios given.	• displayed little ability to interpret or correct code and were unable to write basic algorithms using pseudocode • showed a very limited understanding of OOP that exposed little practical experience of relevant programming • displayed limited knowledge of standard data structures such as stacks and queues. They had some AO1 knowledge, but struggled more with AO2 application • demonstrated very limited knowledge in levels of response questions and struggled to apply their learning to the scenarios given.

Section A overview

Section A of the paper consisted of 8 questions, each with a separate theme.

Question	Topics covered
1	Procedural programming constructs
2	Computational methods – data mining
3	Graphs, A* algorithm and heuristics
4	Queues
5	Stacks
6	Linear search, insertion sort and Big O notation
7	Record structures, binary trees and tree traversal algorithms
8	Abstraction and decomposition

Question 1 (a) (i)

1 A student has written this pseudocode algorithm:

```
01    a = 12
02    do
03        b = input("Enter a number")
04    until b >= 0 and b <= 100
05    for c = 1 to a
06        print(c * a)
07    next c
```

(a) The program uses variables.

(i) Describe what is meant by a variable.

.....

.....

.....

..... **[2]**

Most candidates gained some credit for identifying that a variable holds a value, and this was the most popular response. Element/identifier on its own was insufficient for the first mark point as there had to be a clear indication that the identifier referred to a memory/storage location.

Question 1 (a) (ii)

(ii) Give the identifiers of all the variables used in this program.

..... **[1]**

The majority of candidates had little difficulty correctly identifying the variables given in the code.

Question 1 (b)

(b) The student has used a do loop on line `02`.

Describe the difference between a do loop and a while loop.

.....

.....

.....

.....

[2]

Many responses were vague and did not accurately identify or distinguish that a while loop is a pre-condition loop while a do loop is a post-condition loop. Responses had to be clear as to the relative position of the condition in the loop. Accepted responses included a while loop is a pre-condition loop while a do loop is a post-condition loop. Fewer candidates identified that the body of a while loop may not be executed while the body of a do loop will always be executed at least once.

Question 1 (c)

(c) Rewrite lines 05 to 07 to use a while loop instead of a for loop.

You should write your answer using either program code or pseudocode.

..... **[4]**

This question was generally well attempted with marks relatively evenly distributed throughout the range of marks available. There were some off-by-one errors, with the loop counter or the position of the output line being incorrectly positioned before/after the counter increment. Incorrect loops such as count-controlled for loops were rejected. If responses did not produce a fully correct output, marks given were limited to 3 marks.

Exemplar 1

```
~~while~~ a
c = 1
while c < a :
    print(c*a)
    c = c + 1
```

This response showed a typical off-by-one error where the candidate had not thought through the logic of the entire response. The condition operator should have been <= rather than < to iterate through all values of c from 1 to 12.

Question 2*

2* A company runs a Virtual Learning Environment (VLE). Schools can register students to use the VLE. The students get their own account and the school can view and monitor their students who are registered. There are currently over 10 000 schools registered, each with up to 1000 students.

The students can watch videos, take quizzes and communicate using forums and online chat tools.

The company gathers a large amount of data and wants to use data mining to help them decide how to improve the VLE in the future.

Discuss how the company can use data mining to decide how to improve the VLE.

You should include the following in your answer:

- the characteristics of data mining
- the benefits of data mining in this scenario
- the drawbacks of data mining in this scenario.

[12]

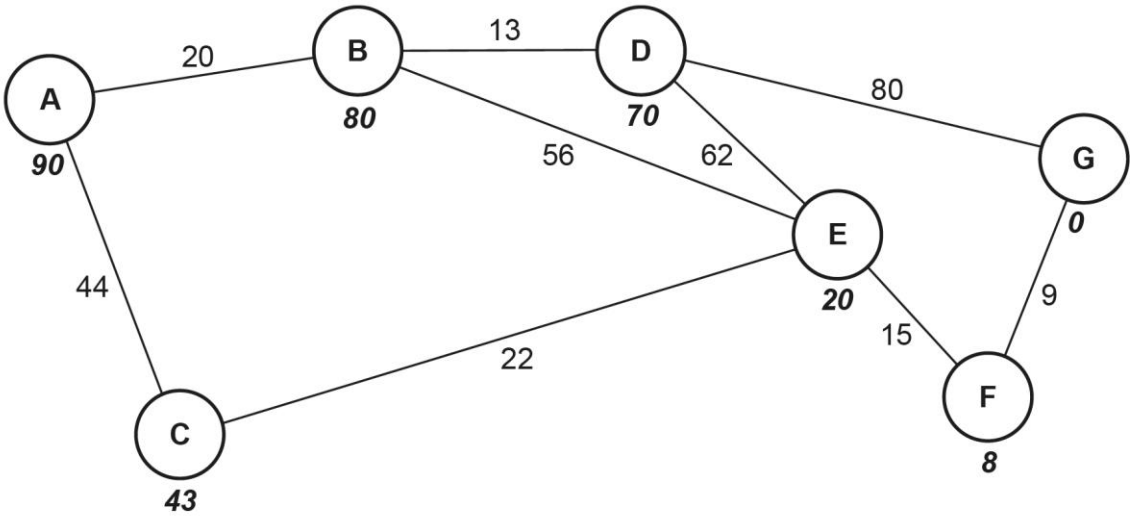
Many Level 2 responses demonstrated an ability to define data mining and give relevant examples of its use. Few candidates were able to give an evaluative critique as to the effectiveness of data mining for this scenario.

Many candidates struggled to structure their responses. Where candidates did produce a clear structure, they often used paragraphs effectively, starting with a definition, considering benefits and drawbacks, then going on to evaluate within the parameters of the scenario.

Question 3 (a)

3 A computer game has a building containing 7 rooms. There are secret passages between each room. **Fig. 3** shows the rooms and the passages between the rooms represented as a graph data structure.

Fig. 3



(a) The number in bold below each node in **Fig. 3** is the heuristic value.

Perform an A* algorithm on the graph shown in **Fig. 3** to find the shortest path from A to G.

Show your working, the nodes visited and the distance.

You may use the table below to give your answer.

.....

.....

.....

Node	Distance travelled	Heuristic	Distance travelled + Heuristic	Previous node

Final path: .....

Distance: .....

[7]

Many candidates demonstrated a good understanding of the A* algorithm and most candidates achieved at least some of the marks available, with the most commonly given being the final path (often through inspection) and the first row of the table.

A few candidates treated the algorithm as if it were Dijkstra's and explored all possible paths/routes rather than stopping as soon as the goal node was located.

Question 3 (b)

(b) State **four** ways that a graph data structure is different from a tree data structure.

1 .....

.....

2 .....

.....

3 .....

.....

4 .....

.....

[4]

Most candidates struggled to score more than one or two marks for this question. Responses had to be clearly mappable to technical terms, but there was evidence of vague language in many cases. Candidates were expected to be able to talk about root nodes, cycles, hierarchy, directed/undirected edges and connected/disconnected nodes. Clear use of technical terms is expected at this level. A common incorrect response was weighted/unweighted. A minimum spanning tree is a subset of a graph that can have weighted edges.

Question 3 (c)*

(c)* The final game will involve multiple computer-controlled characters and interactive elements that make use of artificial intelligence to determine the moves they will make.

The artificial intelligence will use heuristics to determine where the computer-controlled characters will move in the game.

Discuss how heuristics can be used in algorithms.

You should include the following in your answer:

- the purpose of heuristics
- the benefits and drawbacks of heuristics
- the suitability of heuristics in algorithms within a computer game.

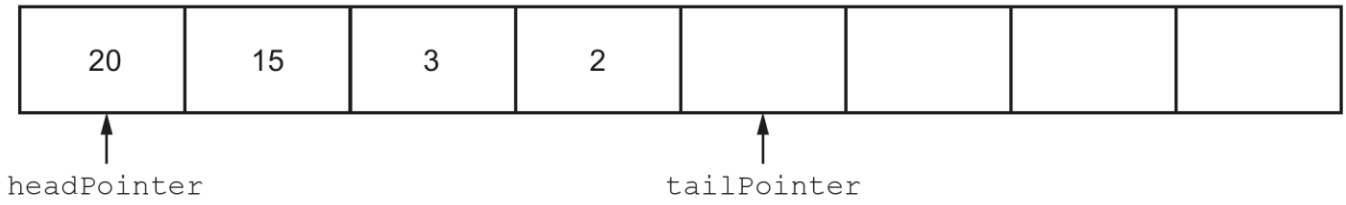
[9]

Candidates demonstrated reasonable knowledge of heuristics in terms of definition and application to the game. Many candidates identified some benefits and drawbacks and the need to have an accurate heuristic value. Few candidates were able to give a detailed evaluation, or expand to scalability for example, to achieve a Level 3 response.

Question 4 (a)

4 The current contents of a queue data structure are shown in **Fig. 4**.

Fig. 4



(a) State the purpose of `headPointer` and `tailPointer` in the queue shown in **Fig. 4**.

`headPointer` .....

.....

`tailPointer` .....

.....

[2]

This question was generally well answered by many candidates. There were some vague responses such as 'start of queue', which was not precise enough to indicate the first item, or the index of the first item, as distinct to the indexes of the actual data structure itself.

Question 4 (b)

(b) `enqueue` will add data to the queue. `dequeue` will remove data from the queue.

Show the contents of the queue and the position of both pointers after the following actions have been executed on the queue shown in **Fig. 4** in the order given:

- `enqueue(20)`
- `dequeue()`
- `dequeue()`

[2]

Most candidates either tended to score full marks or no marks. Many candidates missed labelling the pointers in their diagrams and just gave updated queue values. Some candidates kept the values 20 and 15 on the diagram and correctly moved the `headPointer` to point to index 3 and the `tailPointer` to the next free space available. A considerable number of candidates erroneously shifted all items forward, showing a lack of understanding as to how a queue is efficiently implemented.

Question 4 (c)

(c) The queue is used to store ID numbers of jobs that a program needs to process. Some jobs will be given a priority which means they need to be processed first.

Explain why this queue is **not** a suitable data structure for this program.

.....

.....

.....

..... **[2]**

The first part of many responses was well attempted, with most candidates identifying the First In First Out (FIFO) property of a queue. Far fewer candidates were able to successfully expand on this for the second mark. A linked explanation was required such as a higher priority item cannot be inserted at the front/in the middle of the queue because only the first element can be accessed/dequeued. Some candidates just reiterated what FIFO meant instead of giving the linked explanation to a priority queue which was insufficient to gain the second mark.

Question 5 (a)

5 The contents of a stack are stored in the 1-dimensional array called `numbers`.

`topStack` stores the index of the next free space in the stack.

The array is declared with space for 100 elements.

(a) The function `pop()` returns the next item from the stack and updates the appropriate pointers.

Describe the steps in the function `pop()`.

.....

.....

.....

.....

.....

..... **[4]**

There were many weak responses that did not outline the discrete steps required in the `pop()` function and there was often no reference to the `topStack` pointer.

A number of candidates confused a stack with a queue and talked about head/rear pointers. Some candidates confused 'space for 100 elements' in the question with the top of the stack and then erroneously talked about removing item 100.

Some candidates erroneously stated that the value at `topStack - 1` would be returned before saying that `topStack` would be decremented. The order of the steps was important for the function to operate correctly, and many candidates lost a mark due to this.

Exemplar 2

Describe the steps in the function `pop()`.

if pointer is -1

1. checks if stack is empty, if yes returns an error as nothing to return.
2. ~~goes to~~ goes to topStack - 1 value ~~out~~, return ~~as~~ the value and deletes ~~it~~ data from the pointer
3. moves pointer of topStack to topStack - 1
=> topStack = topStack - 1

This response shows the candidate returning the value from `topStack - 1` in step 2. As soon as the function returns a value no further actions are performed within the function, so decrementing `topStack` in point 3 was not given.

Question 5 (b)

(b) The function `push()` inserts its parameter called `dataValue` onto the stack and updates the appropriate pointers.

Complete the function `push()` using pseudocode or program code.

```
function push(.....)
    if .....!= 100 then
        numbers[.....] = dataValue
        topStack = topStack + .....
        return true
    else
        return false
    endif
endfunction
```

[4]

The majority of candidates scored three or more marks. Some candidates erroneously used `numbers.length` or `len(numbers)` in the second space instead of `topStack`.

Question 5 (c)

(c) Write an algorithm, using pseudocode or program code, to call the function `push()` with the value 15 and output a message saying `"Added"` if the value was successfully inserted onto the stack or `"Not Added"` if the stack is full.

.....

.....

.....

.....

.....

..... **[4]**

Some candidates erroneously tried to make a call such as `Function Push(15)` instead of calling and using/storing the result of `Push(15)`. A number of responses incorrectly used string values "`True`" / "`False`" instead of Boolean `True` / `False`. Many candidates tried to directly access and use the `topStack` pointer instead of using the function return value as required in the question.

Question 6 (a)

6 The following strings are stored in an array.

"rainbow"	"moon"	"sun"	"stars"	"clouds"	"tornado"

(a) Explain how a linear search would search the array for the index that stores "clouds".

.....

.....

.....

.....

.....

..... **[3]**

Most candidates scored the majority of the marks available and demonstrated a clear understanding of a linear search. Many candidates answered by example with values from the given list.

Question 6 (b)

(b) State why a binary search cannot be used in this example.

.....

..... **[1]**

Most candidates correctly identified the requirement for data to be sorted/ordered for a binary search to work. 'Organised' was too vague and was not accepted.

Question 6 (c)

(c) Show how an insertion sort will sort the given data into **ascending** alphabetical order.

"rainbow"	"moon"	"sun"	"stars"	"clouds"	"tornado"

.....

.....

.....

.....

.....

..... **[5]**

Nearly half the candidates achieved full marks and clearly demonstrated the steps involved in an insertion sort for the given data. Some candidates confused insertion sort with either bubble, merge or selection sort, and so scored no marks for not answering the question.

Exemplar 3

(c) Show how an insertion sort will sort the given data into **ascending** alphabetical order.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

"rainbow"	"moon"	"sun"	"stars"	"clouds"	"tornado"

rainbow | moon, sun, stars, clouds, tornado

moon, rainbow | sun, stars, clouds tornado

moon, rainbow, sun | stars, clouds, tomado

moon, rainbow, stars, sun | clouds, tornado

clouds, moon, rainbow, stars, sun | tornado

clouds, moon, rainbow, stars, sun, tomado |

- A sorted list is made with an element in the list. The elements ~~The most~~ in the old list (right) are sorted into the new list (left) in ascending order.

This exemplar demonstrated the state of the dataset after each pass of the insertion sort. The candidate has separated the sorted and unsorted parts of the list to make the response much clearer.

Question 6 (d) (i)

(d)

(i) A sorting algorithm has a best **time** complexity of O(n).

Describe what is meant by the best **time** complexity O(n) for a sorting algorithm.

.....

.....

.....

..... **[2]**

Many candidates did not identify linear but gave a description for 'directly proportional' or vice-versa, so only gained one mark. Candidates needed to specify that the time required increased in direct proportion to the number of items. Proportion on its own was too vague and could have meant inverse proportion for example.

Incorrect responses included 'quickest time to find / least operations' defining best case rather than answering the question regarding O(n).

Question 6 (d) (ii)

(ii) Another sorting algorithm has a worst **space** complexity of O(log(n)).

Describe what is meant by the worst **space** complexity O(log(n)) for a sorting algorithm.

.....

.....

.....

..... **[2]**

Many candidates did not identify logarithmic but tried to give a description or vice-versa. Many candidates just said the memory space increased in proportion to the log of n without explaining what this meant. For the second mark candidates had to explain that as the number of items n increased the amount of additional memory required became progressively smaller.

Question 6 (d) (iii)

(iii) Identify the **time** complexity that means the time will not change even when the number of items increases.

..... **[1]**

The vast majority of candidates gave the correct response of constant or O(1).

Question 6 (d) (iv)

(iv) Identify the **space** complexity that means the amount of memory (space) used will double each time a new item is included.

..... **[1]**

Half the candidates correctly identified exponential complexity, but common erroneous responses included $O(n^2)$ or O(2n) instead of $O(2^n)$.

Misconception

A number of candidates erroneously thought that n^2 or 2^n demonstrated exponential growth instead of 2^n. Candidates need to have the mathematical grounding to understand the difference between different Big O growth factors.

Question 7 (a)

7 A computer game stores tasks that the player has requested. Each task has:

- an identification (ID) number e.g. **Task A**
- a real number to be processed e.g. **123456.789**
- an integer number to represent the order the tasks should be accessed e.g. **1**.

The task that needs to be processed the earliest is given the order number 1.

Two or more tasks can have the same order number. For example, two tasks can have an order number 1.

(a) The data about each task needs to be stored. This will store the ID number, data value and order number for a task.

Explain why a record data structure is suitable for this data.

.....

.....

.....

..... **[2]**

Record structures were poorly understood, and it was clear that many candidates had very limited experience of using records / structures within a programming language. Many candidates gave responses related to database records rather than record data structures.

Question 7 (b) (i)

(b) The tasks will be stored in a binary search tree before they are processed. They are stored in ascending order by their order number.

(i) Give **two** characteristics of a binary search tree.

1 .....

.....

2 .....

.....

[2]

Some candidates gave generic properties of trees such as 'root' instead of specific characteristics of a binary search tree as required by the question. There was then some lack of precision when describing the number of child nodes each parent node could have (maximum two, not always two), or lack of clarity in defining 'ordered' without qualifying what this meant.

Question 7 (b) (ii)

(ii) Give an advantage of storing the tasks in a binary search tree instead of a 1-dimensional array.

.....

..... **[1]**

A number of candidates erroneously talked about binary tree traversal to traverse a 1D list. However, there were many clear and correct responses. Some unqualified 'easier' / 'quicker' type responses gained no credit. Candidates had to identify that it was quicker to search/insert into the tree.

Question 7 (b) (iii)

(iii) Tick (✓) **one** column in each row to identify whether each statement applies to a depth-first (post-order) tree traversal, a breadth-first tree traversal, or neither of these two traversals, when performed on a binary search tree.

Statement	Depth-first (post-order)	Breadth-first	Neither of these two traversals
All nodes at the current depth are visited before moving to the next depth			
The algorithm traverses to the end of one branch before moving to another branch			
The algorithm will make use of backtracking			
The traversal can be used to output the contents of the tree in ascending order			
The algorithm will output the root node last			

[5]

Many candidates achieved at least partial credit if not full credit.

Question 7 (b) (iv)

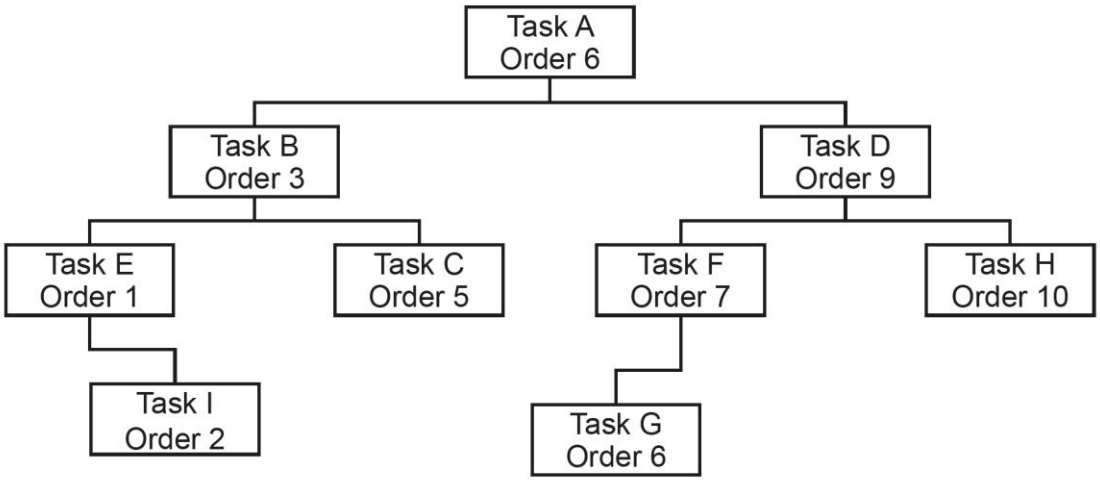
(iv) The tasks currently stored in the binary search tree are shown here.

When a new task is inserted with the same order number as a pre-existing task, it is classed as having a higher order number.

For example, task G has the same order number as task A. Since task G was inserted after task A it is classed as a higher number.

Change the diagram to show the contents of the binary search tree after the following tasks are inserted in the order given:

- Task X with order number 12
- Task Y with order number 7
- Task Z with order number 11



[3]

The binary search tree diagram generally was answered well with the majority of candidates gaining full marks.

Question 8 (a)

8 A group of students are designing a racing car game. The game will allow players to enter their name and then a choice of vehicle. They will then race against other vehicles that will be controlled by the program. Players will use the arrow keys to control their vehicle.

(a) The students are identifying the inputs and outputs for the game.

Complete the table by identifying **two** inputs **and two** outputs for the game.

Input 1	....
Input 2	....
Output 1	....
Output 2	....

[4]

While many candidates scored full marks for this question for identifying suitable inputs and outputs within the context of the problem a number of candidates scored no marks for erroneously giving input or output devices. Sometimes responses were unqualified such as 'left arrow key' rather than 'left arrow to turn left' where candidates did not identify an input within the context of the scenario as an input to the game.

Question 8 (b) (i)

(b) The students use abstraction during the design process.

(i) State what is meant by abstraction **and** describe how it can be used to design the racing car game.

Definition .....

.....

Use .....

.....

.....

.....

[3]

Many candidates scored two marks for defining abstraction and giving one example of its application to the game, but fewer managed to identify a second example for full marks.

Question 8 (b) (ii)

(ii) Explain why it is beneficial to use abstraction when designing a computer program such as a game.

.....

.....

.....

.....

.....

..... **[3]**

Some candidates reiterated a definition of abstraction and did not focus on the benefits of abstraction as the question required. Most candidates struggled to give more than one or two benefits. There were also a number of unqualified responses such as 'saves time' that did not specify 'saving coding development time'.

Question 8 (c) (i)

(c)
(i) The group of students also use decomposition.

State what is meant by decomposition.

.....

..... **[1]**

Most candidates correctly defined decomposition, but some candidates lost marks for lack of clarity for not specifying that a problem is broken down into sub problems.

Question 8 (c) (ii)

(ii) Describe **one** benefit of using decomposition when designing a computer program such as a game.

.....

.....

.....

..... **[2]**

Many candidates identified that decomposition would make it simpler to code individual subproblems but then lacked a clear related expansion. The most popular response was that this allowed workload to be shared between different members of a team to reduce overall development time or to play to team member expertise.

Section B overview

Section B referred to one specific scenario that revolved around a problem that was solved by using Object Oriented Programming (OOP) techniques. Those candidates who had very limited practical experience of OOP often struggled with the questions in section B. Conversely, those candidates who demonstrated strong coding skills often scored very well.

Assessment for learning

The OOP scenario in Section B lends itself to implementation by candidates as a programming exercise to enhance the learning of associated concepts such as constructor methods, instantiation and class methods.

Question 9 (a)

9 A game is being written that makes use of object-oriented programming. A prototype for one part of the game is being designed that includes a character, a road and a prize to collect.

The road will have 50 spaces that a character can move along. Each space on the road will store a null value or a prize object for the user to collect. Each space is numbered sequentially from the first space (position 0) to the last space (position 49) and will not change during the game. As the player travels down the road, the position the player is on the road will be output.

(a) The road is designed to be a 1-dimensional array with the identifier `road`.

Explain why an array is a suitable data structure to represent the road.

.....

.....

.....

.....

.....

..... **[3]**

Many responses were too vague, showing little knowledge of the properties of arrays. Relatively few candidates appeared to be able to make explicit links to the scenario to achieve full marks.

Question 9 (b) (i)

(b) The characters and prizes are designed as separate classes. 10 of the spaces on the road will contain an instance of the class `Prize`. The other spaces will be empty.

The class design for `Prize` is here.

class: Prize
attributes: private name : string private type : string private value : integer
methods: new() getName() getType() getValue()

`new()` is the constructor method. The name, type and value are passed to the constructor as parameters which then assigns these to the attributes.

(i) The method `getName()` returns the data in the attribute `name`.

Write the method `getName()` using pseudocode or program code.

.....

.....

.....

..... **[2]**

While many candidates had little difficulty giving code for a `getter()` there were a number of common errors. Some candidates used a private access modifier when a `getter()` needs to be public. There was often erroneous use of 'procedure' whereas a `getter()` is a function that must return a value. Some candidates tried to set values within the `getter()` function when it should only have returned the class attribute value.

Question 9 (b) (ii)

(ii) A global 1-dimensional array, `allPrizes`, stores 10 objects of type `Prize`.

The prize in index 3 has the name "Box", the type is "money" and the value is 25.

Write pseudocode or program code to create a new object for this prize and store it in index 3 of `allPrizes`.

.....

.....

.....

..... **[3]**

Many candidates struggled with the instantiation of an object. Where candidates made an attempt to instantiate some did not use a string for "box" and "money" or did not give 25 as an integer but instead gave the string "25".

Question 9 (b) (iii)

(iii) The game starts with 10 prizes. Each prize is allocated to one space on the road.

An algorithm needs designing that will generate a random space on the road for each prize. Each road space can only store one prize.

Describe the decisions that will need to be made in this algorithm and how these will affect the program flow.

.....

.....

.....

.....

.....

..... **[3]**

There were only two reasonable decisions that could be given from the scenario details. Candidates needed to make it clear that a decision with a Boolean output was present that would dictate two potential outcomes. Some candidates quoted actions such as 'randomly assign space for prize' which did not represent a decision. Many responses described the mechanics of setting up the game and the random spaces but did not highlight the program conditions/decisions as required.

Question 9 (c) (i)

(c) The class design for `Character` is here.

class: Character
attributes: private name : string private money : integer private experience : integer private roadPosition : integer
methods: new() getName() getMoney() getExperience() getRoadPosition() changePosition() updateValues()

The four get methods return the associated attribute.

The number of moves is passed to `changePosition()` as a parameter. The method adds this value to the character's position on the road.

The type and value of an object are passed to `updateValues()` as parameters. If the object is money the value is added to the character's money. If the type is experience the value is added to experience. If the type is neither money or experience no changes are made.

(i) `new()` is the constructor method. The name of the character is passed into the constructor as a parameter. The constructor then initialises both the experience and road position of the character to 0 and initialises the amount of money to 5.

Write the constructor method for `Character` using either pseudocode or program code.

You do not need to declare the class, the attributes or any other methods.

.....

.....

.....

.....

.....

..... **[5]**

It was clear that those candidates with limited OOP programming knowledge found the writing of a relatively simple constructor method difficult. Those with relevant programming experience often found this to be a very straightforward question. Common errors included passing additional values to set the `experience`, `roadPosition` and `money` attributes rather than setting them to the constant values indicated in the question.

Question 9 (c) (ii)

(ii) The type and value of a prize are passed as parameters to the method `updateValues`. If the type is money the value is added to the character's money. If the type is experience then the value is added to the experience. If the type is neither money or experience no changes are made.

For example, for the Character `player1`:

`player1.updateValues("money",10)` updates player1's money by 10

`player1.updateValues("experience",5)` updates player1's experience by 5

`player1.updateValues("foo",9)` has no effect on player1.

Write pseudocode or program code for the method `updateValues()`.

.....

.....

.....

.....

.....

..... **[5]**

The `updateValues` procedure again proved problematic for candidates with limited OOP experience. No marks were given for the first mark point if a function was declared as there was no return value. Parameter names needed to be fit for purpose, understandable, and had to match the order given in the question scenario to work for the given example calls.

Question 9 (d)

(d) This incomplete pseudocode algorithm:

- creates a new character with the name Jamal
- loops until the character reaches the end of the road
- generates a random number of spaces to move between 1 and 4 (including 1 and 4)
- moves the character and checks if the new space has a prize
- updates the character attributes if there is a prize
- outputs the character's new attribute values.

Complete the pseudocode algorithm.

```
character1 = new …… ("Jamal")

newPosition = 0

while newPosition < ……

    move = random(1, 4) //this will generate a random number between 1 and 4

    character1.changePosition(move)

    newPosition = character1.getRoadPosition()

    if newPosition < 50 and road[……] != null then

        prizeType = road[newPosition].getType()

        valueAmount = road[newPosition].getValue()

        character1.updateValues(……, valueAmount)

        print("Congratulations you are in position", newPosition, "and found",

             road[newPosition].getName())

        print("Money =", character1.getMoney(), "and experience =",

             character1. …… ())

    endif

……

print("You reached the end of the road")
```

[6]

Nearly all candidates achieved some marks, and a majority scored five or six marks.

Question 9 (e)

(e) The procedure `displayRoad()` outputs the contents of each space in the road. The number of each space is output with either:

- the word "empty" if there is no prize
- the name of the prize if there is a prize.

```
01    procedure displayRoad()

02        for x = 0 to 60

03            print("Space", y)

04            if road[x] == null then

05                print("empty")

06            elseif

07                print(road[x].getValue())

08            endif

09        next x

10    endprocedure
```

The algorithm contains errors.

Give the line number of **four** different errors and write the corrected line for each error.

Error 1

Error line 1 .....

Correction .....

Error 2

Error line 2 .....

Correction .....

Error 3

Error line 3 .....

Correction .....

Error 4

Error line 4 .....

Correction .....

[4]

Many candidates scored three or four marks but in general candidates found it harder to identify errors in the code than to complete code in the previous question. Some candidates didn't give the line number but rewrote the incorrect line before giving the corrected line, which was acceptable, although not ideal given the scaffolding.

Question 9 (f)*

(f)* A programmer is going to create a prototype for one small part of the game. Both `road` and `allPrizes` will be needed throughout the whole prototype. The programmer is considering making these global arrays as she thinks it will reduce the development time. Another programmer has suggested that doing this may create some problems when the rest of the game is created at a later stage.

Compare the use of global and local variables in this program.

You should include the following in your answer:

- the use of local and global variables
- alternative methods to using global variables
- the appropriateness of each to this program design.

[9]

Most responses were Level 2 for definitions and some expansion to passing parameters. Very few candidates were able to go into depth about alternatives to global variables such as passing by value and passing by reference in detail or extending to issues such as scalability within a larger more extended game. Few candidates picked up on the fact that this was a more limited prototype that was likely to be expanded on which would require more consideration to be given to variable scope.

Supporting you

Teach Cambridge

Make sure you visit our secure website **Teach Cambridge** to find the full range of resources and support for the subjects you teach. This includes secure materials such as set assignments and exemplars, online and on-demand training.

Don't have access? If your school or college teaches any OCR qualifications, please contact your exams officer. You can forward them this link to help get you started.

Reviews of marking

If any of your students' results are not as expected, you may wish to consider one of our post-results services. For full information about the options available visit the OCR website.

Access to Scripts

We've made it easier for Exams Officers to download copies of your candidates' completed papers or 'scripts'. Your centre can use these scripts to decide whether to request a review of marking and to support teaching and learning.

Our free, on-demand service, Access to Scripts is available via our single sign-on service, My Cambridge. Step-by-step instructions are on our website.

Keep up-to-date

We send a monthly bulletin to tell you about important updates. You can also sign up for your subject specific updates. If you haven't already, sign up here.

OCR Professional Development

Attend one of our popular professional development courses to hear directly from a senior assessor or drop in to a Q&A session. Most of our courses are delivered live via an online platform, so you can attend from any location.

Please find details for all our courses for your subject on **Teach Cambridge**. You'll also find links to our online courses on NEA marking and support.

Signed up for ExamBuilder?

ExamBuilder is a free test-building platform, providing unlimited users exclusively for staff at OCR centres with an Interchange account.

Choose from a large bank of questions to build personalised tests and custom mark schemes, with the option to add custom cover pages to simulate real examinations. You can also edit and download complete past papers.

Find out more.

Active Results

Review students' exam performance with our free online results analysis tool. It is available for all GCSEs, AS and A Levels and Cambridge Nationals (examined units only).

Find out more.

You will need an Interchange account to access our digital products. If you do not have an Interchange account please contact your centre administrator (usually the Exams Officer) to request a username, or nominate an existing Interchange user in your department.

Online courses

Enhance your skills and confidence in internal assessment

What are our online courses?

Our online courses are self-paced eLearning courses designed to help you deliver, mark and administer internal assessment for our qualifications. They are suitable for both new and experienced teachers who want to refresh their knowledge and practice.

Why should you use our online courses?

With these online courses you will:

- learn about the key principles and processes of internal assessment and standardisation
- gain a deeper understanding of the marking criteria and how to apply them consistently and accurately
- see examples of student work with commentary and feedback from OCR moderators
- have the opportunity to practise marking and compare your judgements with those of OCR moderators
- receive instant feedback and guidance on your marking and standardisation skills
- be able to track your progress and achievements through the courses.

How can you access our online courses?

Access courses from Teach Cambridge. Teach Cambridge is our secure teacher website, where you'll find all teacher support for your subject.

If you already have a Teach Cambridge account, you'll find available courses for your subject under Assessment - NEA/Coursework - Online courses. Click on the blue arrow to start the course.

If you don't have a Teach Cambridge account yet, ask your exams officer to set you up – just send them this link and ask them to add you as a Teacher.

Access the courses **anytime, anywhere and at your own pace**. You can also revisit the courses as many times as you need.

Which courses are available?

There are **two types** of online course: an **introductory module** and **subject-specific** courses.

The introductory module, Building your Confidence in Internal Assessment, is designed for all teachers who are involved in internal assessment for our qualifications. It covers the following topics:

- the purpose and benefits of internal assessment
- the roles and responsibilities of teachers, assessors, internal verifiers and moderators
- the principles and methods of standardisation
- the best practices for collecting, storing and submitting evidence
- the common issues and challenges in internal assessment and how to avoid them.

The subject-specific courses are tailored for each qualification that has non-exam assessment (NEA) units, except for AS Level and Entry Level. They cover the following topics:

- the structure and content of the NEA units
- the assessment objectives and marking criteria for the NEA units
- examples of student work with commentary and feedback for the NEA units
- interactive marking practice and feedback for the NEA units.

We are also developing courses for some of the examined units, which will be available soon.

How can you get support and feedback?

If you have any queries, please contact our Customer Support Centre on 01223 553998 or email support@ocr.org.uk.

We welcome your feedback and suggestions on how to improve the online courses and make them more useful and relevant for you. You can share your views by completing the evaluation form at the end of each course.

Need to get in touch?

If you ever have any questions about OCR qualifications or services (including administration, logistics and teaching) please feel free to get in touch with our customer support centre.

Call us on
01223 553998

Alternatively, you can email us on
support@ocr.org.uk

For more information visit
ocr.org.uk/qualifications/resource-finder
ocr.org.uk
facebook.com/ocrexams
twitter.com/ocrexams
instagram.com/ocrexaminations
linkedin.com/company/ocr
youtube.com/ocrexams

We really value your feedback

Click to send us an autogenerated email about this resource. Add comments if you want to. Let us know how we can improve this resource or what else you need. Your email address will not be used or shared for any marketing purposes.

I like this **I dislike this**

Please note – web links are correct at date of publication but other websites may change over time. If you have any problems with a link you may want to navigate to that organisation's website for a direct search.

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT