Oxford Cambridge and RSA

GCE

Computer Science

Advanced **H446**

OCR Report to Centres June 2018

About this Examiner Report to Centres

This report on the 2018 Summer assessments aims to highlight:

- areas where students were more successful
- main areas where students may need additional support and some reflection
- points of advice for future examinations

It is intended to be constructive and informative and to promote better understanding of the specification content, of the operation of the scheme of assessment and of the application of assessment criteria.

Reports should be read in conjunction with the published question papers and mark schemes for the examination.

The report also includes links and brief information on:

- A reminder of our **post-results services** including **reviews of results**
- Link to **grade boundaries**
- **Further support that you can expect from OCR**, such as our Active Results service and CPD programme

Reviews of results

If any of your students' results are not as expected you may wish to consider one of our reviews of results services. For full information about the options available visit the OCR website. If University places are at stake you may wish to consider priority service 2 reviews of marking which have an earlier deadline to ensure your reviews are processed in time for university applications: http://www.ocr.org.uk/administration/stage-5-post-results-services/enquiries-about-results/service-2-priority-service-2-2a-2b/

Grade boundaries

Grade boundaries for this, and all other assessments, can be found on Interchange.

Further support from OCR

activeresults

Active Results offers a unique perspective on results data and greater opportunities to understand students' performance.

It allows you to:

- Review reports on the **performance of individual candidates**, cohorts of students and whole centres
- **Analyse results** at question and/or topic level
- **Compare your centre** with OCR national averages or similar OCR centres.
- Identify areas of the curriculum where students excel or struggle and help **pinpoint strengths and weaknesses** of students and teaching departments.

http://www.ocr.org.uk/administration/support-and-tools/active-results/getting-started/

CPD Hub
Your route to OCR's teacher training

Attend one of our popular CPD courses to hear exam feedback directly from a senior assessors or drop in to an online Q&A session.

https://www.cpdhub.ocr.org.uk

CONTENTS

Advanced GCE

Computer Science

(H446)

OCR REPORT TO CENTRES

H446/01 Computing principles

1. General Comments:

In general, candidate responses demonstrated subject knowledge appropriate to the specification. The examination was evidently accessible to candidates and clearly differentiated effectively. The majority of candidates were prepared for the rigour of the examination.

Some candidates again, as in previous series, found questions challenging when they were required to write HTML, SQL and programming statements. With some candidates unable to write basic code. Centres must encourage candidates to practise these skills.

Too many candidates failed to gain credit when asked to demonstrate their knowledge of basic definitions simply because of poor clarity of expression.

Centres should encourage candidates to apply their knowledge to the question when required. Too many candidates lost marks for failing to respond in context.

2. Comments on Individual Questions:

Qu.	Comment
1a(i)	The majority of candidates answered this question well although some candidates stated that 'instructions and data are stored in the same memory location' more attention to detail is required at this level of study.
1a(ii)	A whole range of features were accepted for this question. Most candidates stated an appropriate feature but some then did not go on to describe how the feature improved performance.
1b(i)	Most candidates achieved the first mark on this question. The second mark was lost by those who ticked 'red light' for an input of 5.
1b(ii)	Many candidates gave a comprehensive description of the fetch execute cycle but did not apply their response to the assembly code instruction given in the question therefore losing marks.
1b(iii)	Most candidates gained credit for writing code to read the value from the user with some then losing marks for either specifying the incorrect condition e.g. IF value < 5 and/or not outputting the result.
1b(iv)	Candidates were assessed on the quality of their extended response in this question. Most candidates could describe the basic differences between assembly code and high level languages, with many giving examples of where each would be best used. Some candidates gave clear and appropriate justification for the coffee machine being programmed in assembly code. In general, most candidates scored reasonably well on this question.

Qu.	Comment
2a(i)	This question enabled candidates to demonstrate basic knowledge of the functions of an operating system, many scored well but some candidates responses were too generic for this level of study e.g. 'manage resources'.
2a(ii)	Most candidates correctly identified paging.
2a(iii)	Most candidates clearly described virtual memory but some did not go on to discuss the movement of pages between memory and virtual memory. More so discussing the movement of whole programs in and out of virtual memory.
2b(i)	Surprisingly fewer candidates than anticipated gained full marks on this question. Many candidates gained some marks. Marks were invariably lost on the HTML for the hyperlink.
2b(ii)	This question was poorly attempted. Many candidates mentioned a crawler/spider program but then failed to clearly describe the process. Some candidates went on to discuss ranking even though the question specifically stated not to.
2b(iii)	Most candidates achieved two marks on this question with few referring to the fact that RISC requires less complex circuitry.
3a	Many candidates achieved full marks on this question. Candidates were credited for describing the statement and/or stating the output. Those who did not achieve full marks generally stated that the statement 'outputs the flight numbers of flights with the destination of JFK' omitting to state that the flight numbers will be extracted from the flight table.
3b	Most candidates did not achieve the first mark for the DELETE statement because they included the wildcard i.e. DELETE *. Many went on to achieve the rest of the marks giving the criteria, using correct SQL statements.
3c(i)	Virtually all candidates correctly identified the lack of uniqueness of DestinationCode as the reason why it should not be the primary key.
3c(ii)	Some candidates clearly did not know what a secondary key is, describing a foreign key instead.
3d(i)	The definition for 1NF was answered well by most candidates.
3d(ii)	The definition for 2NF was not as well answered with most candidates omitting to state that the database must first be in 1NF.
3d(iii)	Fewer candidates scored well on this part of the normalisation question. Many identified that there was a transitive relationship between DestinationName and DestinationCode but few could describe this with clarity.
3e	Many candidates gave vague responses to this question and therefore did not gain credit. Candidates should be encouraged to learn specific formats/methods for exchanging data (specification reference 1.3.2b) some examples could include: CSV; API. A whole range of formats/methods were accepted for this question.
4a	Although most candidates stated that a WAN is a wide area network, many did not go on to state that devices on a WAN are connected over a large geographical area.
4b(i)	Most candidates explained what a protocol is rather than why they are important on a network. Candidates should be encouraged to apply their knowledge to the question being asked. Some candidates gained credit for correctly explaining that protocols enable devices to interpret data in the same way, allowing them to communicate.
4b(ii)	Most candidates achieved three or four marks on this question, the layers were accepted in any order. Those who did not, invariably scored zero marks.
5a	Despite the question, specifically outlining the limitations of user mobility some candidates still responded with hardware devices that require use of your hands. Some also gave the technique as opposed to the hardware e.g. speech recognition rather than microphone.

Qu.	Comment
5b (i) (ii)	Invariably, all candidates fared well on both parts of this question.
5c	Most candidates stated that open source software was free but some went on to explain that users can amend/inspect the code rather than source code. Again, attention to detail is required at this level of study.
6	Candidates were assessed on the quality of their extended response in this question. Many candidates offered a balanced discussion although some were limited in scope i.e. limiting the discussion to one law. Few conclusions were appropriately justified/reasoned leading to many candidates scoring in the mid-level band on this question.
7a	There were three marks available for this question. Candidates should be reminded that they need to specify a clear and specific point for each mark awarded. Many candidates did not extend their responses to three points. In addition, there were many definitions of the form 'the car needs to be able to respond in real time' which gained no credit.
7b(i)	In this question, it was evident that more candidates have had practical experience of programming in an Object Oriented Programming languages than previous series but there were still many who clearly have not. Those candidates who are practised in programming using OOP answered well here.
7b(ii)	Many candidates clearly described encapsulation but not in context, as the question asked, therefore not gaining credit.
7b(iii)	Most candidates stated an advantage of using encapsulation with many then repeating their first statement. Therefore achieving one mark.
7c	It was evident in responses to this question that some candidates did not read the question carefully. The methods to be called in their response were given in the question. The variable to be checked was passed as a parameter. Candidates who used this information correctly scored well on this question.
7d	There were a range of acceptable responses to this question. Some candidates did not gain credit because the advantage/disadvantage stated was not specific to the customer as the question asked. Candidates should be reminded to read the question carefully.
8a 8b	Most candidates scored well on the first two parts of this question.
8c	Few candidates scored more than two marks on this question. In most cases, this was due to the fact that the code overwrote the original array value when shifting. Only the top scoring candidates appreciated that an intermediate temporary variable/array was required to hold the original value(s).
8d	Candidates were assessed on the quality of their extended response in this question. Many candidates explained the difference between symmetric and asymmetric encryption very well. Some went on to describe the circumstances in which they are used equally well. Few discussed the impact on society. Many candidates scored in the mid band on this question.
9a	Most candidates achieved both marks on this question.
9b	Responses which used binary subtraction or two's compliment were accepted for this question. Most candidates scored two marks.
9c	Most candidates achieved both marks on this question.
9d	Most candidates achieved full marks on this question. Those who did not, invariably scored zero marks.

Qu.	Comment
9e	Surprisingly, this question was poorly attempted. Few candidates gained full marks with many not gaining any credit. Candidates should be reminded that the first two bits of the mantissa must be different in normalised floating point numbers.
9f 9g	Invariably, all candidates achieved the available mark on each of these question parts.
10a	Most candidates achieved both the available marks on this question.
10b	The question required candidates to find the Boolean expression represented in the Karnaugh Map. Most candidates achieved a mark for showing the correct groupings on the map. Many went on to achieve some marks for the resultant expression. Some candidates specified NOT(C AND D) instead of NOT C AND NOT D evidently assuming the expressions are the same.

H446/02 Algorithms and programming

1. General Comments:

The paper differentiated the candidates effectively and scripts included some very strong candidate responses.

Questions that targeted Knowledge and Understanding required candidates to have studied the whole specification and to have learnt the relevant definitions. Some candidates had not been prepared by covering the whole specification and thus failed to achieve marking points targeted at lower grades for basic recall e.g. binary search.

Questions targeting Application' required higher order skills to be able to use knowledge gained in context to solve problems. There was clear differentiation between candidates who understood the concepts and who could apply them, and those who displayed little ability to apply what they had learnt. It was very clear that a lot of candidates have had very limited exposure to Object Oriented Programming (OOP) and have not written programs that require classes with attributes and methods to be defined. Section B of the exam paper will always have a scenario that uses OOP and candidates do need experience of writing code in this paradigm to do well.

Certain types of question benefit greatly from being answered by using clear diagrams. The questions on Dijkstra's algorithm, the trace of a recursive function and the demonstration of an insertion sort were all questions that were best answered through the use of clear diagrams and tables with appropriate annotation. Where possible, when a computation algorithm is being executed, the use of verbose text should be discouraged.

2. Comments on Individual Questions:

Qu.	Comment
1a	The majority of candidates had little difficulty with this question, but a surprising number of errors were made because candidates could not sort alphabetically past the first letter of the data values given to be inserted into the tree.
1b	Most candidates found this to be straight-forward. Where there was confusion, candidates often performed a depth-first traversal, instead of a breadth-first traversal.
1c(i)	Some candidates found the first two marks more difficult to access than the last three, because they did not fully understand that the parameters to the function were to be used.
1c(ii)	Many candidates recognised that a binary search tree was required, but some lost marks when they specified that 2 children were always required for each parent node, when it is a maximum of two children that are allowed, so that there can be 0, 1 or 2 children.
2a(i)	Candidates found it easier to define decomposition than recognition. There were many circular definitions of the form problem recognition means recognising the problem that were not creditworthy.
2a(ii)	The most common answer that candidates gave was abstraction, but a wide range of valid responses that included the named computational methods in the specification were seen.

Qu.	Comment
2b(i)	Whilst many candidates had some familiarity with the term data mining there were a number of very vague answers. Candidates need to know that very large volumes of data are collected with the intention of finding patterns and trends.
2b(ii)	Many candidates could give a good response that explained how data trends identified could be applied in a sales scenario, but some candidates only gave answers related to stock control which was not relevant.
2c(i) & (ii)	Many candidates had clearly not come across the term 'performance modelling' which is in the specification, and they struggled to relate it to the context given. However, some very good responses were given, that included the use of Big O notation analysis and mathematical modelling to determine algorithmic performance.
2d	Whilst many candidates could identify that reusable components would save time in future projects, fewer could identify specific reasons why this would be the case.
3a(i)	Many candidates scored well, but there was evidence of a lack of learning of technical vocabulary e.g. general descriptions of loops rather than cycles in graphs. Candidates are expected to be able to use the correct technical vocabulary such as weighted / directed in reference to graph structures.
3a(ii)	Most candidates scored poorly on this question because they did not contextualise their answers, and did not explain how the graph could be an abstraction of the puzzle in the scenario.
3a(iii)	Few candidates achieved both marks. Common answers that were creditworthy recognised the fact that humans find visualisations of problems easier to understand.
3b	Whilst many candidates were familiar with the calculations involved in Dijkstra's algorithm, most confined their calculations to the shortest path ABEIJ and length 14. It was far less common for candidates to recognise that Dijkstra's algorithm does not stop until all nodes have been visited and that some nodes are over-ridden with lower values. Some candidates simply stated the shortest route without any explanation or calculation and this was not creditworthy. Candidates should be encouraged to produce clear answers in tabular format.
3c	Candidates were well prepared and had obviously used both Dijkstra's algorithm and the A* algorithm, and could explain differences between them and give examples of possible heuristics. Very few candidates could evaluate and discuss how the scaling of the algorithms was relevant to access the top level mark band.
3d	Many candidates did not read the question clearly and consequently did not specify IDE features that were specifically relevant to debugging the program and thus discussed editing features. Good answers included break points, variable watches and stepping.
4a	A significant number of candidates struggled to clearly present a program trace for a recursive algorithm. Those who used diagrammatic or clearly indented structures showing the recursive levels and the return values faired best. Candidates need to be encouraged to work on producing a logical layout for a recursive function trace.
4b	Many candidates either defined passing by value or passing by reference and did not answer the question. Few could demonstrate a deeper understanding of the implications of the method of parameter passing chosen within the context of a recursive function.
4c	Most candidates produced responses limited to the scope of global variables being accessible throughout the program or a discussion of the different methods of parameter passing available. Few made any references to either recursive functions or to the implications to memory usage of using parameters instead of global variables.

Qu.	Comment
4d	Candidates need to demonstrate technical knowledge. Those who answered well produced good descriptions of how a stack is used when a function is called recursively. Few candidates then went on to explain how iteration overwrites the same variables with new values.
5a	The vast majority of candidates had no trouble executing a sequence of push/pop instructions successfully.
5b(i)	A significant number of candidates did not describe a conditional decision clearly and lost marks when merely describing push/pop operations. Where the first part of the question was answered well some candidates then failed to see the second part of the question and did not describe an impact of the condition. Candidates need to be reminded to read and analyse the wording of the whole question to access all marking points.
5b(ii)	Most candidates answered well, but few gave answers that demonstrated an ability to combine separate logical statements with the OR operator. A significant number of candidates used a .lower() method being mostly familiar with Python syntax and methods.
5c	Those candidates with experience of languages other than Python appreciated that a 1D array is a static structure that needs to be declared with a given size, and that a stack pointer variable would be required. It is of concern that significant numbers of candidates have only had experience of lists and their associated methods in Python. A number of candidates also confused their descriptions with those for a queue rather than a stack.
5d(i)	Candidates should be encouraged to demonstrate sorting algorithms through the use of clear diagrams that show the steps/passes in the sorting algorithm. Where candidates used verbose text, it often made it far harder to follow whether or not the correct sorting algorithm had been applied. Most candidates did implement an insertion sort, but some did describe bubble or merge sorts instead.
5d(ii)	The paper title is 'Algorithms and programming'. Binary search is a standard algorithm that should be fully understood by candidates, and candidates should be able to program it. Many candidates produced very vague descriptions that were far too general to credit at this level. Candidates needed to be able to discuss how the upper and lower bound pointers are used in this algorithm (or equivalent for recursive solutions).
5d(iii)	It was a little disappointing to see a number of candidates using variations on for loops rather than rewriting the code using a while loop as required. A significant number of candidates still struggled to demonstrate a coherent logical response that would work for something that is relatively simplistic, thus showing a lack of proficiency in coding practice.
6a(i)	The concept of a record structure in a programming language was poorly understood – with many candidates interpreting it as a database record structure. Those who did score well had a far better understanding of OOP.
6a(ii)	Most candidates answered well, but weaker candidates put examples for the Currency and Date fields rather than creating suitable variable names.
6a(iii)	The structure type was not well understood by many candidates, who used "recordidentifier" as the record identifier, rather than appreciating that *box1* could be assigned the data type *items*. Some candidates incorrectly omitted the quotation marks around the string data or added quotation marks to the numerical data fields.
6b(i)	Most candidates had learned the definition for *queue* and answered successfully.

Qu.	Comment
6b(ii)	Many candidates termed their answers by restating terms in the question. Whilst the term encapsulation was often cited it was less often explained well – few candidates knew that getter() and setter() methods could be used to access the private attributes of a class. Some candidates stated that private attributes could not be changed which demonstrated a clear lack of understanding of the paradigm.
6b(iii)	Few candidates appeared to have had practical experience of programming in an Object Oriented Programming (OOP) languages. Those that did, answered well. It is advisable to ensure that candidates are prepared for section B of the paper by having implemented programs using the OOP methodology.
6b(iv)	Candidates struggled to apply all the information given in the stem of question 6. The stem defined the class *itemQueue* with the private attributes and methods it held. Candidates who did not use this information were unable to produce correct solutions. Only a few of the strong candidates realised that it was a circular queue that was implemented, and therefore checked the increment of the tail pointer.
6b(v)	Lack of practical experience and lack of correct solutions for 6b(iii) often led to incorrect answers for this question.
6b(vi)	Few candidates realised that you cannot use a single input statement to read in four data items, and only stronger candidates realised that they should use the enqueue method created in b(iv) to add the inputted items into the queue.
6b(vii)	Most candidates answered this well, realising that the data in the queue had to be written and retrieved from secondary storage.
6c	Most candidates seem to know a little bit about each of the techniques of caching and concurrency, but rarely went beyond basic points. Most candidates identified cache with cache memory rather than disk caching of recently retrieved records. Stronger candidates did access the higher mark bands by contextualising their answers, explaining how concurrent processes or parallel algorithms could be applied to searching large volumes of data.

H446/03 Programming project

1. General Comments:

It was pleasing to see an increase in electronic submissions (either using the Repository or CD/ DVD / USB stick by post). These tend to be easier for the moderator to process, cheaper for the centre to post and much better environmentally.

Because projects generally tend to be large (100+ pages) moderators very much appreciate where centres take the effort to help them understand where in the documentation evidence can be found. URS forms with comments explaining why marks were given and page numbers showing where evidence can be found are particularly helpful in ensuring nothing gets overlooked. Similarly, well-structured projects in a single document, with contents pages and subheadings are much easier to navigate.

Most candidates produced projects with a sensible level of challenge, with the potential to demonstrate all the criteria. Centres should bear in mind that when unsure of the suitability of a project they can check it with the subject specialists using the email address ComputerScience@ocr.org.uk

A few centres had candidates all submitting very similarly themed projects. This is strongly discouraged. Candidates need the freedom to be able to demonstrate an independent approach. This is difficult to achieve when a cohort are all working on similar tasks. Instead, candidates should aim to research a task of their choosing.

There was an increase in the use of Unity this year. The Unity engine undoubtedly allows candidates to create impressive projects. Candidates must keep in mind that credit in the implementation is for the code they write. They needn't spend much time describing how they set the project up in Unity other than to set the scene for the reader. Focus should be on the code they have written. (Candidates should also note that credit can only be given for code and not any functionality derived from visual tools such as FlowCanvas)

There were a number of instances this session where projects were submitted with screenshots of code that was unreadable. Care must be taken that code shown in screenshots is big enough and of a high enough resolution. An alternative is to post the code directly in as text. Ideally, syntax highlighting should be preserved. If it is not retained by directly copying from the IDE/text editor there are a number of only prettify systems that may be able to help. It is the responsibility of candidates to ensure that their final pdf or printed copy is readable. Credit cannot be given for code that cannot be read.

2. Comments on Individual Questions:

It should be emphasised that whilst the sections below are listed in the order of the mark scheme, may candidates successfully structured their project in a non-linear fashion.

Analysis

Some candidates tended to give lengthy descriptions of what computational methods were. This should be discouraged as it is of no benefit to them. Their focus should be on why their project is suited to be solved by a computer not a regurgitation of the theory covered in Paper 2.

It is recognised that some projects lend themselves more to discussion of hardware and software requirements and as such some candidates will have very little to write on this aspect.

The requirements and success criteria are part of the key to a good project. Candidates who produced vague requirements tended to perform less well in the later sections than those who came up with a more prescriptive set. The lowest scoring candidates had very vague requirements “Must be easy to use.” “Must load quickly” that could have applied to most systems. An ideal set of requirements should be detailed enough to pass onto a third party to design the system.

Design

Candidates who had opted to take an object-oriented approach tended to fare particularly well in the design section, having a clear idea as to how their project can be decomposed.

There was still a tendency for some candidates to try and reverse engineer their pseudocode which reduces the credit they can be given.

Developing the Coded Solution

This tended to be the most polarising of the sections.

Some candidates are producing superb accounts of how projects were developed. The show each what they intend to accomplish at each iteration and how they coded it. They include plenty of detail about meaningful problems they overcame. (Candidates who find they are not encountering non-trivial problems are probably not taking on projects with sufficient challenge.)

At the lower end, candidates were just providing copies of their code with brief commentaries.

A small number of centres had successfully used version control software to help candidates keep records of iterations. This was an excellent idea and had the added benefit of exposing them to using VCS. Candidates need to ensure that their repositories are not publicly viewable whilst their coursework is live.

Testing to Inform Development

In the best instances of this section, the majority of focus was on failed tests. Candidates took time to elaborate on tests that initially failed and how they remedied them.

Testing to Inform Evaluation

In both the testing sections, candidates need to ensure they provide sufficient evidence of tests. Evidence need not be exhaustive for every test, but by end of this section, the reader should be left in no doubt the system functions in the manner claimed.

There was an increase in evidence for this section being provided by means of video files. This works particularly well with visual programs such as games where it is hard to infer if a test has been successful from a screenshot.

Evaluation

A number of candidates struggled with cross-referencing their objectives with their tests. Some simply asserted how successful they felt they had been for each objective. Candidates must relate their success back to the testing they have completed. It is still possible to get full marks in this section without having met all requirements providing the candidate has given evidence of those they have met and made meaningful reflections on those they were not able to meet.

About OCR

OCR (Oxford Cambridge and RSA) is a leading UK awarding body. We provide qualifications which engage people of all ages and abilities at school, college, in work or through part-time learning programmes.

As a not-for-profit organisation, OCR's core purpose is to develop and deliver general and vocational qualifications which equip learners with the knowledge and skills they need for their future, helping them achieve their full potential.

OCR (Oxford Cambridge and RSA Examinations)
The Triangle Building
Shaftesbury Road
Cambridge
CB2 8EA

OCR Customer Contact Centre

Telephone: 01223 553998
Facsimile: 01223 552627
Email: general.qualifications@ocr.org.uk

www.ocr.org.uk

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored

Oxford Cambridge and RSA Examinations
is a Company Limited by Guarantee
Registered in England
Registered Office:
The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA
Registered Company Number: 3484466
OCR is an exempt Charity

OCR (Oxford Cambridge and RSA Examinations)
Head office
Telephone: 01223 552552
Facsimile: 01223 552553

Cambridge Assessment

CERTIFIED ISO 9001 UKAS MANAGEMENT SYSTEMS 001