OCR
Oxford Cambridge and RSA

AS Level Computer Science
H046/02 Algorithms and problem solving
Sample Question Paper

Date – Morning/Afternoon
Time allowed: 1 hour 15 minutes

Do not use:
- a calculator

First name										
Last name										
Centre number						**Candidate number**				

INSTRUCTIONS
- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided.
- Additional paper may be used if required but you must clearly show your candidate number, centre number and question number(s).
- Do **not** write in the bar codes.

INFORMATION
- The total mark for this paper is **70**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **12** pages.

Answer **all** questions.

1 (a) Describe what is meant by the term *IDE (Integrated Development Environment)*.

……

……

……**[2]**

(b) Identify and describe **three** features commonly found in an IDE that will help programmers to find any bugs in their code.

1……..

……

……

2……..

……

……

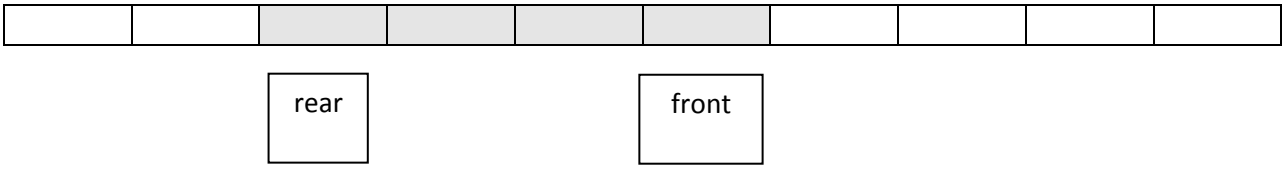
3……..

……

……**[6]**

2 The array queue shown below is set up to hold a small queue. Assume that there is sufficient storage to hold all necessary additions to the queue.

Queue



The table below shows variables that are used to maintain the queue:

Variable	Type	Purpose
front	integer	pointer to the front element of the queue
rear	integer	pointer to the rear element of the queue
queue_full	Boolean	indicates whether the queue is full
max	integer	the maximum size of the queue

Shown below is an algorithm that is intended to add an item to the queue.

```
procedure add_to_queue (item)

    if ((front - rear) +1) == max then
      queue_full=true

    else
       front=front + 1
       queue[front]=item

     endif

endprocedure
```

(a) Identify the parameter that is passed to this procedure.

……**[1]**

(b) Describe the logical decision that is made.

……

……

……**[2]**

(c) (i) This algorithm contains a logic mistake. Explain what the mistake is.

.....

.....

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

(ii) Rewrite the algorithm to correct the mistake.

.....

.....

.....

.....

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

3 (a) Describe **one** difference between a global and a local variable.

……

……

……**[2]**

(b) Explain, using **one** example, why global variables are sometimes used in a program.

……

……

……

……

……

……**[3]**

(c) Explain why good programming practice generally avoids the use of global variables.

……

……

……

……

……

……**[2]**

(d) Explain why parameter passing to a function can be a better alternative to using global variables.

……

……

……

……**[2]**

4 Nobugs is a software development company that produces enterprise-wide management software for large companies. Its software products are built up from many program functions.

The managers of Nobugs enforce standard rules on their programmers about how program functions should be written.

The following are some of the rules that they insist upon:

- no function may be longer than a single page of code
- variable identifiers must conform to a standard convention
- each function must have a single entry point
- variables must not be set up outside the scope of a function
- hardware-specific code must be avoided
- embedded documentation must be adequate.

(a) Describe what is meant by a function.

.....

.....

.....

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

(b) Compare a program procedure with a function.

.....

.....

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

(c) A programmer at Nobugs has written some program code that includes two user defined functions.

```
function my_function1(number)
    return number*number*number
endfunction

function my_function2(number)
    return number*number*number*number
endfunction

number=int(input("Enter a number "))
print(my_function1(number))
print(my_function2(number))
```

(i) Apart from the two functions written by the programmer, identify **two** other functions used in this piece of program code.

1……..

……

……

2……..

……

……**[2]**

(ii) The programmer tests this code with the input value of 3. State the output that would be obtained.

……

……

……**[2]**

(d)* Nobugs enforces standard rules about writing functions on its programmers. Discuss the reasons why this might be the case.

……

……

……

……

……

……

……

……

……

……

……

……**[9]**

5 A car has a feature in its engine management system that is intended to save fuel and emissions produced when the car is waiting at traffic lights or in a traffic jam. The default option is that if the gears are disengaged and the car is not moving, the engine is switched off. There is a display on the dashboard that indicates when the engine has been switched off in this way.

However, sometimes it is necessary to keep the engine running even when the car is stationary, in order to provide electric power to charge the battery, run the heater, run the air conditioning system or keep the lights on. This, in turn, is affected by the external and internal temperatures, the settings chosen by the driver and the intensity of light outside.

(a) Identify **four** inputs needed by this feature of the engine management system.

For each one suggest a suitable data type for its storage.

Input	Data type

[8]

(b) Identify **two** outputs from this engine management feature.

……

……

……**[2]**

(c) Write an algorithm that would produce the results described for this engine management system feature.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....**[8]**

6 DriveSim Tutor is a 3D driving simulator program designed to allow learner drivers to practice following the Highway Code whilst driving through a virtual town.

The simulator's developers study a real town. They then use abstraction on their findings before designing a virtual town.

(a) Explain why it is necessary for the developers to use abstraction.

……

……

……**[2]**

(b) As a result of abstraction there will be similarities and differences between the virtual and real town.

(i) State **two** similarities there might be between the virtual and real town. Explain why these similarities exist.

1……..

……

……

2……..

……

……**[4]**

(ii) State **two** differences between the virtual and real town.

1……..

……

2……..

……**[2]**

(c) A road in the town has a “no overtaking” sign.

Describe how the simulator would check the driver obeys this sign whilst on this road.

.....

.....

.....

.....

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

OCR
Oxford Cambridge and RSA

…day June 2014 – Morning/Afternoon

AS Level Computer Science

H046/02 Algorithms and problem solving

SAMPLE MARK SCHEME

Duration: 1 hour 15 minutes

MAXIMUM MARK 70

DRAFT

This document consists of 16 pages

MARKING INSTRUCTIONS

PREPARATION FOR MARKING

SCORIS

1. Make sure that you have accessed and completed the relevant training packages for on-screen marking: *scoris assessor Online Training*; *OCR Essential Guide to Marking*.

2. Make sure that you have read and understood the mark scheme and the question paper for this unit. These are posted on the RM Cambridge Assessment Support Portal http://www.rm.com/support/ca

3. Log-in to scoris and mark the **required number** of practice responses ("scripts") and the **required number** of standardisation responses.

   YOU MUST MARK 10 PRACTICE AND 10 STANDARDISATION RESPONSES BEFORE YOU CAN BE APPROVED TO MARK LIVE SCRIPTS.

TRADITIONAL

Before the Standardisation meeting you must mark at least 10 scripts from several centres. For this preliminary marking you should use **pencil** and follow the **mark scheme**. Bring these **marked scripts** to the meeting.

MARKING

1. Mark strictly to the mark scheme.

2. Marks awarded must relate directly to the marking criteria.

3. The schedule of dates is very important. It is essential that you meet the scoris 50% and 100% (traditional 50% Batch 1 and 100% Batch 2) deadlines. If you experience problems, you must contact your Team Leader (Supervisor) without delay.

4. If you are in any doubt about applying the mark scheme, consult your Team Leader by telephone, email or via the scoris messaging system.

5. Work crossed out:
   a. where a candidate crosses out an answer and provides an alternative response, the crossed out response is not marked and gains no marks
   b. if a candidate crosses out an answer to a whole question and makes no second attempt, and if the inclusion of the answer does not cause a rubric infringement, the assessor should attempt to mark the crossed out answer and award marks appropriately.

6. Always check the pages (and additional objects if present) at the end of the response in case any answers have been continued there. If the candidate has continued an answer there then add a tick to confirm that the work has been seen.

7. There is a NR (No Response) option. Award NR (No Response)
   - if there is nothing written at all in the answer space
   - OR if there is a comment which does not in any way relate to the question (e.g. 'can't do', 'don't know')
   - OR if there is a mark (e.g. a dash, a question mark) which isn't an attempt at the question.

   Note: Award 0 marks – for an attempt that earns no credit (including copying out the question).

8. The scoris **comments box** is used by your Team Leader to explain the marking of the practice responses. Please refer to these comments when checking your practice responses. **Do not use the comments box for any other reason.**
   If you have any questions or comments for your Team Leader, use telephone, email or the scoris messaging system.

9. Assistant Examiners will send a brief report on the performance of candidates to their Team Leader (Supervisor) via email by the end of the marking period. The report should contain notes on particular strengths displayed as well as common errors or weaknesses. Constructive criticism of the question paper/mark scheme is also appreciated.

10. **Annotations**

Annotation	Meaning

11. Subject-specific Marking Instructions

INTRODUCTION

Your first task as an Examiner is to become thoroughly familiar with the material on which the examination depends. This material includes:

- the specification, especially the assessment objectives
- the question paper and its rubrics
- the mark scheme.

You should ensure that you have copies of these materials.

You should ensure also that you are familiar with the administrative procedures related to the marking process. These are set out in the OCR booklet **Instructions for Examiners**. If you are examining for the first time, please read carefully **Appendix 5 Introduction to Script Marking: Notes for New Examiners**.

Please ask for help or guidance whenever you need it. Your first point of contact is your Team Leader.

USING THE MARK SCHEME

Please study this Mark Scheme carefully. The Mark Scheme is an integral part of the process that begins with the setting of the question paper and ends with the awarding of grades. Question papers and Mark Schemes are developed in association with each other so that issues of differentiation and positive achievement can be addressed from the very start.

This Mark Scheme is a working document; it is not exhaustive; it does not provide 'correct' answers. The Mark Scheme can only provide 'best guesses' about how the question will work out, and it is subject to revision after we have looked at a wide range of scripts.

The Examiners' Standardisation Meeting will ensure that the Mark Scheme covers the range of candidates' responses to the questions, and that all Examiners understand and apply the Mark Scheme in the same way. The Mark Scheme will be discussed and amended at the meeting, and administrative procedures will be confirmed. Co-ordination scripts will be issued at the meeting to exemplify aspects of candidates' responses and achievements; the co-ordination scripts then become part of this Mark Scheme.

Before the Standardisation Meeting, you should read and mark in pencil a number of scripts, in order to gain an impression of the range of responses and achievement that may be expected.

In your marking, you will encounter valid responses which are not covered by the Mark Scheme: these responses must be credited. You will encounter answers which fall outside the 'target range' of Bands for the paper which you are marking. Please mark these answers according to the marking criteria.

Please read carefully all the scripts in your allocation and make every effort to look positively for achievement throughout the ability range. Always be prepared to use the full range of marks.

LEVELS OF RESPONSE QUESTIONS:

The indicative content indicates the expected parameters for candidates' answers, but be prepared to recognise and credit unexpected approaches where they show relevance.

Using 'best-fit', decide first which set of BAND DESCRIPTORS best describes the overall quality of the answer. Once the band is located, adjust the mark concentrating on features of the answer which make it stronger or weaker following the guidelines for refinement.

- **Highest mark:** If clear evidence of all the qualities in the band descriptors is shown, the HIGHEST Mark should be awarded.
- **Lowest mark:** If the answer shows the candidate to be borderline (i.e. they have achieved all the qualities of the bands below and show limited evidence of meeting the criteria of the band in question) the LOWEST mark should be awarded.
- **Middle mark:** This mark should be used for candidates who are secure in the band. They are not 'borderline' but they have only achieved some of the qualities in the band descriptors.

Be prepared to use the full range of marks. Do not reserve (e.g.) high Band 3 marks 'in case' something turns up of a quality you have not yet seen. If an answer gives clear evidence of the qualities described in the band descriptors, reward appropriately.

	AO1	**AO2**	**AO3**
High (thorough)	Precision in the use of question terminology. Knowledge shown is consistent and well-developed. Clear appreciation of the question from a range of different perspectives making extensive use of acquired knowledge and understanding.	Knowledge and understanding shown is consistently applied to context enabling a logical and sustained argument to develop. Examples used enhance rather than detract from response.	Concerted effort is made to consider all aspects of a system/problem or weigh up both sides to an argument before forming an overall conclusion. Judgements made are based on appropriate and concise arguments that have been developed in response resulting in them being both supported and realistic.
Middle (reasonable)	Awareness of the meaning of the terms in the question. Knowledge is sound and effectively demonstrated. Demands of question understood although at times opportunities to make use of acquired knowledge and understanding not always taken.	Knowledge and understanding applied to context. Whilst clear evidence that an argument builds and develops through response there are times when opportunities are missed to use an example or relate an aspect of knowledge or understanding to the context provided.	There is a reasonable attempt to reach a conclusion considering aspects of a system/problem or weighing up both sides of an argument. However the impact of the conclusion is often lessened by a lack of supported judgements which accompany it. This inability to build on and develop lines of argument as developed in the response can detract from the overall quality of the response.

Low (basic)	Confusion and inability to deconstruct terminology as used in the question. Knowledge partial and superficial. Focus on question narrow and often one-dimensional.	Inability to apply knowledge and understanding in any sustained way to context resulting in tenuous and unsupported statements being made. Examples if used are for the most part irrelevant and unsubstantiated.	Little or no attempt to prioritise or weigh up factors during course of answer. Conclusion is often dislocated from response and any judgements lack substance due in part to the basic level of argument that has been demonstrated throughout response.

	Assessment Objective
AO1	Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.
AO.1	Demonstrate **knowledge** of the principles and concepts of abstraction, logic, algorithms, data representation or other as appropriate.
AO1.2	Demonstrate **understanding** of the principles and concepts of abstraction, logic, algorithms, data representation or other as appropriate.
AO2	Apply knowledge and understanding of the principles and concepts of computer science including to analyse problems in computational terms.
AO2.1	Apply knowledge and understanding of the principles and concepts of computer science.
AO2.2	Analyse problems in computational terms.
AO3	Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.
AO3.1	Design computer systems that solve problems.
AO3.2	Program computer systems that solve problems.
AO3.3	Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.

Question			Answer	Marks	Guidance
1	(a)		• A (single) program (1) used for developing programs (1) made from a number of components (1).	**2 AO1.1 (2)**	Up to 2 marks for a valid description.
	(b)		• Debugging tools allow inspection of variable values (1 – AO 1.1) this can allow run-time detection of errors (1 – AO1.2). • Code can be examined as it is running (1 – AO1.1) which allows logical errors to be pinpointed (1 – AO1.2). • IDE debugging can produce a crash dump (1 – AO1.1), which shows the state of variables at the point where an error occurs (1 – AO1.2). • It can display stack contents (1 – AO1.1) which show the sequencing through procedures/modules (1 – AO1.2). • It can step through code (1 – AO1.1), which allows the programmer to watch the effects each line of code (1 – AO1.2). • The insertion of a break-point (1 – AO1.1) allows the program to be stopped at a predetermined point in order to inspect its state (1 – AO1.2).	**6 AO1.1 (3) AO1.2 (3)**	1 mark (AO1.1) for each correct identification up to a maximum of three identifications plus up to a further 1 mark (AO1.2) for each of three valid descriptions.
2	(a)		• Item (1).	**1 AO2.1 (1)**	For 1 mark.
	(b)		• The queue is checked to determine if it is full (1), if it is, then stop (1) otherwise continue (1).	**2 AO2.1 (2)**	Up to 2 marks for valid description that demonstrates application of knowledge and understanding to given context.
	(c)	(i)	• The front of the queue is incremented (1), an item is placed at (near) the front of the queue (1), it should be placed at the rear (1).	**2 AO2.2 (2)**	Up to 2 marks for a valid explanation.
		(ii)	`front=front + 1` changed to `rear=rear-1` (1) `queue[front]=item` changed to `queue[rear]=item` (1)	**2 AO3.2 (2)**	For 2 marks.

Question			Answer	Marks	Guidance
			rear=rear-1 queue[rear]=item		
3	(a)		• Global variable is visible throughout a program / may be accessed from more than one part of the program (1), local variable is visible only in module/construct where it is created/declared (1).	**2** **AO1.2** **(2)**	Up to 2 marks for a valid description.
	(b)		• A global variable used where a value needs to be accessible from various parts of a program (1 – AO1.2), it is the same value (1 – AO1.2). • Irrespective of the place where it is accessed (1 – AO1.2). • Any sensible example of a value that will reasonably need to be the same, e.g. today's date, VAT rate, pi (1 – AO2.1).	**3** **AO1.2** **(2)** **AO2.1** **(1)**	Up to 2 marks (AO1.2) for a valid explanation. Up to 1 mark (AO2.1) for a valid example.
	(c)		• Global variables make it difficult to integrate modules (1), they increase complexity of a program (1), they may cause conflicts with names written by others/in other modules (1), they may be changed inadvertently when program is complex (1).	**2** **AO1.2** **(2)**	Up to 2 marks for a valid explanation.
	(d)		• When a parameter is passed to a function, only a <u>copy</u> of the data passed may be changed (1), therefore no unforeseen effects will occur in other modules (1). This is the case only if passed by value (1).	**2** **AO1.2** **(2)**	Up to 2 marks for a valid explanation.
4	(a)		• A function is a named section of program (1) that performs a specific task (1). • It returns a value (1), it is often called inline (1).	**2** **AO1.1** **(2)**	Up to 2 marks for a valid description.
	(b)		• A procedure is also a sub-section of a program (that performs a specific task) (1) but it does not necessarily return a value (1). Most programming languages nowadays	**2** **AO1.2** **(2)**	Up to 2 marks for a valid comparison.

Question			Answer	Marks	Guidance
			use functions (1).		
	(c)	(i)	• int (1). • input (1). • print (1).	**2 AO2.1 (2)**	Up to 2 marks for valid identifications that demonstrates application of knowledge to given context.
		(ii)	• 27 (1). • 81 (1).	**2 AO2.1 (2)**	Up to 2 marks for stating valid output that demonstrates application of knowledge to given context.
	(d) *		**Mark Band 3–High Level** **(7-9 marks)** The candidate demonstrates thorough knowledge and understanding of reasons why Nobugs enforces standard rules about writing functions on its programmers; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well-balanced. Evaluative comments are consistently relevant and well-considered. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2-Mid Level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of reasons why Nobugs enforces standard rules about writing functions on its programmers; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly	**9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)**	**AO1: Knowledge and Understanding** The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • No function may be longer than a single page of code: this is to reduce complexity and aid readability. • Variable identifiers must conform to a standard convention: this helps others to understand the code and reduces the likelihood of duplication, makes maintenance easier. • Each function must have a single entry point: this reduces complexity and makes the search for any bugs more straightforward. • Variables must not be set up outside the scope of a function: this sets a limit on where to look for bugs and reduces the likelihood of a problem spread across many modules. **AO2.1: Application** The selected knowledge/examples should be directly related to the specific question. The following is indicative of possible

Question	Answer	Marks	Guidance
	relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of reasons why an organisation enforces standard rules about writing functions on its programmers with limited understanding shown; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • Explanation of how the standard rules for programming would impact upon the choices made for using functions and variables and how they are addressed. • Discussion around the use of different functions and variables that are dependent, independent or interdependent. **AO3.3: Evaluation** Candidates will need to consider a variety of viewpoints in relation to following standard rules for functions and variables while developing management software and will make evaluative comments about the issues and solutions they are discussing e.g. • Why using functions longer than one page of code will increase complexity? • Why hardware-specific code must be avoided? • Why variables must not be setup outside the scope of a function? • How a single entry point reduces complexity and makes the search for any bugs more straightforward? • What will happen if embedded documentation is not adequate?

Question			Answer	Marks	Guidance
5	(a)		• Target temperature (1 – AO2.1) integer/floating point (1 – AO3.1). • Wheel movement (1 – AO2.1) Boolean (1 – AO3.1). • Engine running (1 – AO 2.1) Boolean (1 – AO3.1). • Internal temperature (1 – AO2.1) integer/floating point (1 – AO3.1). • External temperature (1 – AO2.1) integer/floating point (1 – AO 3.1). • External light level (1 – AO2.1) integer/floating point (1 – AO3.1). • Heating on (1 – AO2.1) Boolean (1 – AO3.1). • Air conditioning on (1 – AO 2.1) Boolean (1 – AO3.1). • Gears engaged (1 – AO2.1) Boolean (1 – AO3.1).	**8** **AO2.1** **(4)** **AO3.1** **(4)**	Up to 4 marks (AO2.1) one mark for each correct identification of input. Up to 4 marks (AO3.1) one mark for identifying the correct data type. Any example of driver choices/settings related to something switched on (1 – AO2.1) Boolean (1 – AO3.1). Any example of driver choices/settings related to a level being set (1 – AO2.1) integer/floating point (1 – AO3.1).
	(b)		• Start engine (1), stop engine (1), signal to dashboard display (1).	**2** **AO3.1** **(2)**	1 mark for each correct identification up to a maximum of two identifications.
	(c)		• Use of endless loop while ignition is turned on (1). • If external temp<target then override (1). • If external light<target then override (1). • If internal temperature<set temperature and heater on then override (1). • If internal temperature>set temperature and air conditioning on then override (1). • If battery output<target then override (1). • If wheels not moving and gears not engaged and not override then stop engine (1). • If override and engine not running then start engine (1).	**8** **AO3.2** **(8)**	Up to 8 marks - 1 mark for each correct step in process. Various approaches could be correctly used. Allow credit for evidence of each step as given, however expressed.

Question			Answer	Marks	Guidance
6	(a)		• A real town contains things that aren't relevant to the simulation (1) which would require unnecessary programming/design effort (1). • …would require extra computational resources… (1). • …could detract from the main purpose of the program (1).	**2 AO2.2 (2)**	Up to 2 marks for a valid explanation.
	b	(i)	• Road signs/road markings (1) – so the user can practise obeying these when driving (1). • Traffic Lights (1) – so user can practise obeying traffic light signals (1). • Zebra crossing (1) – so user can practise slowing down/stopping at zebra crossing (1). • Cars/vehicles (1) - so user can practice driving with other cars on the road (1). • Pedestrians (1) – so user can practice looking out for and avoiding pedestrians (1).	**4 AO2.2 (4)**	1 mark for each correct identification up to a maximum of two identifications plus up to a further 1 mark for each of two valid explanations.
	b	(ii)	• Scenery may be simplified (1). • Smaller roads may be removed (1). • Potholes may be removed (1). • Buildings may be simplified (1). • Imperfections/wear/damage in road markings and signs will be ignored (1). • No need to worry about sounds of real town (1).	**2 AO2.2 (2)**	1 mark for each correct identification up to a maximum of two identifications.
	(c)		• Have a Boolean variable for if user has obeyed this rule (1) and set it to true (1). • Check if on the no overtaking section of road (1). • When entering this section make note of the car immediately in front (1). • If the position of this car becomes behind the user set a flag that this rule has been broken (1)/alert the user that rule has been broken (1). • Continue checking until left the overtaking section (1).	**3 AO2.2 (3)**	Up to 3 marks for a valid description.

Assessment Objectives (AO) Grid

Question	AO1.1	AO1.2	AO2.1	AO2.2	AO3.1	AO3.2	AO3.3	Total
1(a)	2	0	0	0	0	0	0	**2**
1(b)	3	3	0	0	0	0	0	**6**
2(a)	0	0	1	0	0	0	0	**1**
2(b)	0	0	2	0	0	0	0	**2**
2(c)(i)	0	0	0	2	0	0	0	**2**
2(c)(ii)	0	0	0	0	0	2	0	**2**
3(a)	0	2	0	0	0	0	0	**2**
3(b)	0	2	1	0	0	0	0	**3**
3(c)	0	2	0	0	0	0	0	**2**
3(d)	0	2	0	0	0	0	0	**2**
4(a)	2	0	0	0	0	0	0	**2**
4(b)	0	2	0	0	0	0	0	**2**
4(c)(i)	0	0	2	0	0	0	0	**2**
4(c)(ii)	0	0	2	0	0	0	0	**2**
4(d)*	2	2	2	0	0	0	3	**9**
5(a)	0	0	4	0	4	0	0	**8**
5(b)	0	0	0	0	2	0	0	**2**
5(c)	0	0	0	0	0	8	0	**8**
6(a)	0	0	0	2	0	0	0	**2**
6(b)(i)	0	0	0	4	0	0	0	**4**
6(b)(ii)	0	0	0	2	0	0	0	**2**
6(c)	0	0	0	3	0	0	0	**3**
Totals	**9**	**15**	**14**	**13**	**6**	**10**	**3**	**70**

*** = extended response**

BLANK PAGE