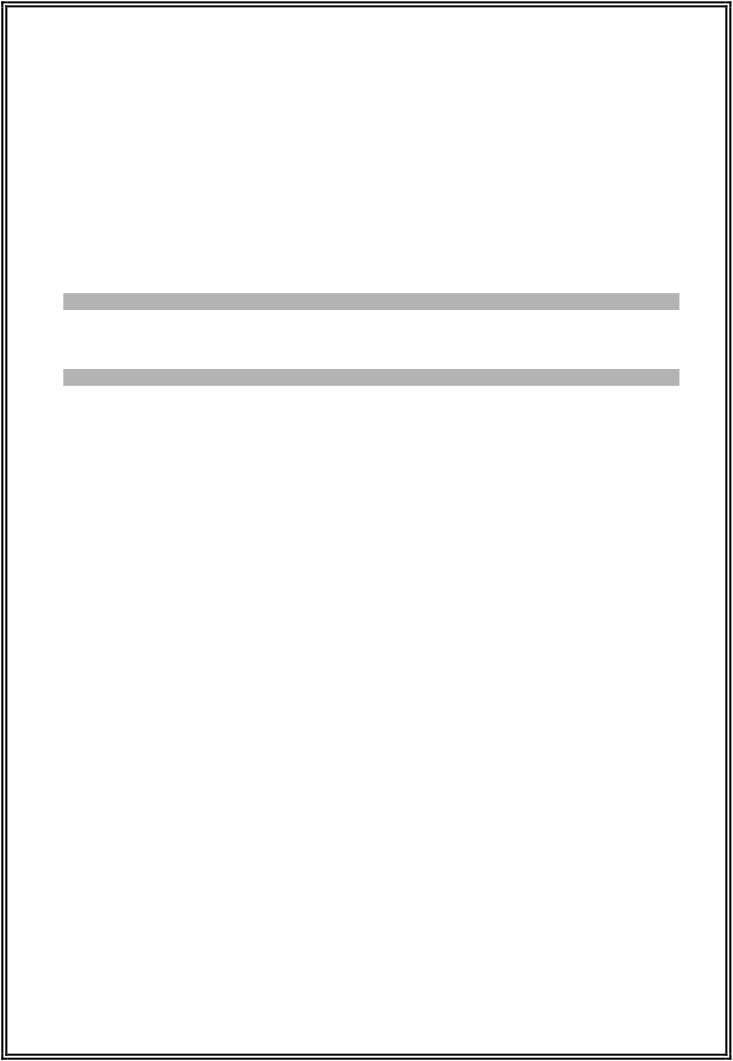
**Evaluation Only. Created with Aspose.Words. Copyright 2003-2023 Aspose Pty Ltd.**



**GCE A LEVEL MARKING SCHEME**



**SUMMER 2018**

**A LEVEL (NEW)**

**COMUTER SCIENCE - UNIT 3 1500U30-1**

**INTRODUCTION**

This marking scheme was used by WJEC for the 2018 examination. It was finalised after detailed discussion at examiners' conferences by all the examiners involved in the assessment. The conference was held shortly after the paper was taken so that reference could be made to the full range of candidates' responses, with photocopied scripts forming the basis of discussion. The aim of the conference was to ensure that the marking scheme was interpreted and applied in the same way by all examiners.

It is hoped that this information will be of assistance to centres but it is recognised at the same time that, without the benefit of participation in the examiners' conference, teachers may have different views on certain matters of detail or interpretation.

WJEC regrets that it cannot enter into any discussion or correspondence about this marking scheme.

**WJEC GCE A Level Computer Science - Unit 3 Mark Scheme Summer 2018**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qu | Answer | Mark | AO1 | AO2 | AO3 | Total |
| 1a | Award 1 mark root node Award 1 mark correct structure | 1 1 |  | 2a 2b |  | 2 |
| 1b | G D B A C F E L I J O M  One mark for each of the following up to a maximum of two  Clone a tree  Count the number of leaves  Convert expression tree to prefix notation | 1 1 | 1b | 2a |  | 2 |
| 1c | A B C D E F G I J L M O  Sort/search a binary tree Accept traversing alphabetically | 1 1 | 1b | 2a |  | 2 |
| 1d | A C B E F D J I M O L G  One mark for each of the following up to a maximum of two  Deleting / Undo a binary tree  Stack-based programming  Convert postfix notation to expression tree | 1 1 | 1b | 2a |  | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qu | Answer | Mark | AO1 | AO2 | AO3 | Total |
| 2a | Functionality – the system must produce correct results for a given set of inputs.  Performance – the system must produce results within an acceptable timeframe. | 1 1 | 1b 1b |  |  | 2 |
| 2b | A natural user interface uses relies on intuitive actions related to natural, everyday human behavior.  One mark for each of the following examples up to a maximum of two marks  Examples include:   * Touch screens, where uses touch or tap graphic icons. * Gesture recognition systems which track and translate user movements into instructions. * Speech recognition systems that identify spoken words and phrases and convert them into instructions.   An immersive interface places one or more of the user’s sense into a computer generated virtual environment.  One mark for each of the following examples up to a maximum of two marks  Examples include:   * Virtual reality headsets or HMDs (head mounted displays) which receive video from a computer, possibly with head tracking (up and down movement). * Binaural or 3D earphones to filter out natural sound and replace with a chosen selected audio. * Force feedback and touch controls provide sensation of using hands within a virtual environment. | 1  1 1  1  1 1 | 1a  1b 1b  1a  1b 1b |  |  | 6 |
| 3 | 1 mark for correct column A + B  1 mark for correct column and column 1 mark for correct column . | 1 1 1 |  | 2a 2a 2a |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qu | Answer | Mark | AO1 | AO2 | AO3 | Total |
| 4a | Award one mark for each of the following:  A.B + A  Using De Morgan’s Law A.B = A + B  A + B + A  Using Boolean identity A + A = 1  B + 1  Using Boolean identity B + 1 **=1** | 1 1 1 |  | 2a 2a 2a |  | 3 |
| 4b | Award one mark for each of the following:  A.B.(B +C) + B.C + B  A.B. B + A.B.C + B.C + B A.0 +A.B.C + B(C + 1)  (B. B = 0)  A.B.C + B  (C + 1 = 1)  B(A.C + 1)  B  (A.C + 1 = 1) | 1 1 1 1 1 |  | 2a 2a 2a 2a 2a |  | 5 |
| 5a | Award two marks for each of the following, one for feature, one for description, up to a maximum of 4:  Auto completion or code completion  Suggests or completes the function being typed including variables and arguments  OR  Bracket matching  Useful when coding in a language that uses blocks of code contained within brackets, for detecting missing brackets.  OR  Syntax checks  Recognises and highlights errors in syntax during code input. Maximum 4 marks. 2 marks for naming tools, 2 marks for expansions.  OR  Formatting e.g. indentation or colour coding of variables | 1 1 1 1  1 1  1 | 1b |  |  | 4 |
| 5b | • Converting the source code written by the programmer into machine code / executable code. | 1 | 1b |  |  |  |
| 5c | * Errors in code syntax / syntax errors will prevent translation. * e.g. spelling mistakes in command works / incorrect punctuation. * Logical errors / semantic errors / runtime errors. * e.g. 2 + 2 = 4 included as 2 \* 2 = 8, any error in logic. * divide by 0, infinite loops, referencing missing files.   Maximum 4 marks. 2 marks for naming errors, 2 marks for correct examples. | 1 1  1 1 | 1b |  |  | 4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qu | Answer | Mark | AO1 | AO2 | AO3 | Total |
| 6a | Award 1 mark for order of answer  Award one mark for correct answer (10001001) - | 1 1 |  |

**This document was truncated here because it was created in the Evaluation Mode.**