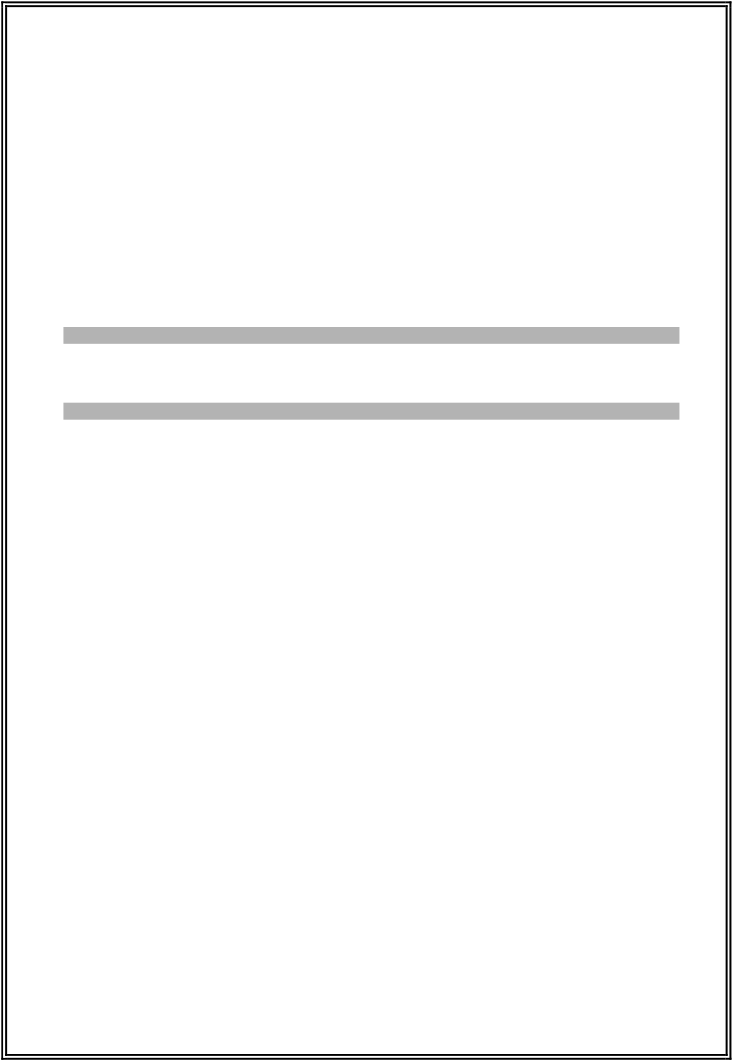
**Evaluation Only. Created with Aspose.Words. Copyright 2003-2023 Aspose Pty Ltd.**



**GCE A LEVEL MARKING SCHEME**



**SUMMER 2017**

**A LEVEL (NEW)**

**COMPUTER SCIENCE - UNIT 4 1500U40-1**

**INTRODUCTION**

This marking scheme was used by WJEC for the 2017 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.

**GCE A LEVEL COMPUTER SCIENCE SUMMER 2017 MARK SCHEME**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Q** |  | **Mark** | **AO1** | **AO2** | **AO3** | **Total** |
| **1(a)** | **Award one mark for:**  SELECT CustName, FlightNum from CUSTOMER | 1 |  |  | 3.1b | 1 |
| **(b)** | **Award one mark for:**  SELECT \* FROM CUSTOMER WHERE FlightNum  = '370'  Or  SELECT CustNum, CustName, DateOfFlight FROM CUSTOMER WHERE FlightNum='370' | 1 1 |  |  | 3.1b 3.1b | 1 |
| **(c)** | **Award one mark for:**  SELECT CustName FROM CUSTOMER WHERE FlightNum =  **Award one mark for:**  (SELECT FlightNum FROM FLIGHT WHERE Terminal = '1')  Accepted but not expected:  SELECT CustName FROM CUSTOMER JOIN FLIGHT WHERE Terminal = '1' | 1 1  2 |  |  | 3.1b 3.1b  3.1b | 2 |
| **(d)** | **Award 1 mark for table; 1 mark for fields (any suitable field size (or type) acceptable):** CREATE TABLE FREQUENTFLYER ( CustNum Char(5)  Points Char(5)) | 2 |  |  | 3.1b | 2 |
| **(e)** | **Award 1 mark for insert; 1 mark for values inserted:**  INSERT INTO FREQUENTFLYER VALUES ('21328' , '210')  INSERT INTO FREQUENTFLYER VALUES  ('14777' , '300') | 2 |  |  | 3.1b | 2 |
| **2** | **Award 1 mark for each point, up to a maximum of 5:**   * Resilient. A problem in one site will not stop other sites from working. * Security. Staff access can be limited to only their portion of the database. * Network traffic is reduced so reducing bandwidth costs. * A single site database still works even if the connection between sites is temporarily   broken).   * Scaling: If demand increases then it is straight forward to add an extra node to the distributed database. (A large company, the demand will likely increase/decrease when a market changes and the system can be scaled quickly). * High performance: Queries and updates are largely local so the there is no network bottleneck (The queries/updates are likely to be local to the sites, and there will be no local bottlenecking). * Expense: either cheaper or more expensive but has to be properly qualified | 5 | 1.1b |  |  | 5 |
| **Q** |  | **Mark** | **AO1** | **AO2** | **AO3** | **Total** |
| **3(a)** | **Award 1 mark for outputting a valid digit (0-9) Award 1 mark for incrementing / decrementing the output values**  **Award 1 mark for the jump**  **Award 1 mark for a working solution**  CLR  JUMP: LDA 1A  OUT  ADD 1B  STA 1A  LDA 1C  DEC 1B  STA 1C  JGT JUMP  CLR - good practice but not required  for mark  Or any similar solution that works. | 1 1 1 1 |  |  | 3.1b | 4 |
| **(b)** | **Award one mark for each correct command:**  Register R 0111 1000 (after instruction LDR R,  2  1D).  Register S 0100 01102 (after instruction LDR S, 1E). [Both required for one mark]  Register R 0011 1100 (after instruction ASR R)  2  Accumulator 0011 1100 (after instruction LDA R)  2  Accumulator 1000 00102 (after instruction ADD S) Register R 1000 0010 (after Instruction STA R)  2 | 1 1  1 1 1 |  | 2.1a 2.1a  2.1a 2.1a 2.1a |  | 5 |
| **4(a)** | **Award one mark for each correct point** Two or more processors working together to perform a single task.  The task is split into smaller sub-tasks (threads). These tasks are executed simultaneously by all available processors (any task can be processed by any processor. | 1  1 1 | 1.1b  1.1b 1.1b |  |  | 3 |
| **Q** |  | **Mark** | **AO1** | **AO2** | **AO3** | **Total** |
| **(b)** | **Award one mark for each of the following up to a maximum of four:**   * More complex to program * Multiple processors may not be available * Software may not allow parallel processing * Advantages may be minimal due to improved processing speed * Sequential processing may still be important so limit to speed gained * Not all tasks are suitable for parallel processing * May be less efficient overall due to introducing additional functions not needed in an equivalent sequential program. | 4 | 1.1b |  |  | 4 |
| **5(a)** | **Award one mark for each point** Truncation removes the least significant bits (moves it nearer to zero)  In rounding the number is approximated to the nearest whole number/tenth/hundredth etc.  0.10011 truncated to 4 bits 0.100  3.75 rounded to 2 digits 3.8  Any valid example where the outcome is different | 1 1  1 1 | 1.1b 1.1b  1.1b 1.1b |  |  | 4 |
| **(b)** | **Award one mark for:** Rounding. | 1 | 1.1b |  |  | 1 |
| **(c)** | **Award one mark for each description** Absolute error = Original – New  Or  Absolute error = New – Original  Relative error = Absolute error / Original | 1  1 | 1.1b  1.1b |  |  | 2 |

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