**2019 JC 2 H2 Computing Paper 2 Marking Scheme**

|  |  |  |  |
| --- | --- | --- | --- |
| **Section A** | | | |
| **Qn** | **Suggested Solutions:** | | **Marks** |
| a | * Records of details of furniture -- by whom, volume, accessibility * Layout of record of furniture -- by whom, accessibility * Transaction records -- by whom, volume, accessibility * Layout of record of furniture -- by whom, accessibility * Order form for the customer -- by whom, accessibility, accuracy, transposition errors * Layout of Order form -- by whom, accessibility * Filing of Order form -- -- by whom, volume, accessibility * Processes all the orders, records -- by whom, volume, accessibility, time factor * Filling out and sends off paper order forms to the suppliers for all items ordered that day. -- by whom, volume, accessibility, accuracy, transposition errors, time factor * All paper documents easily missed place, messy in updating | | 1 mark for every 2 points |
|  | **Sub – total** | | **4** |
| b | C:\Users\s0132874c\Documents\Pages from oct10dippmreport.jpg | | **Award 1 mark:**  Node key, with explanations  **Award 3 marks:**  Correct PERT chart layout  All dependencies & values shown |
|  | **Sub – total** | | **4** |
| c | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Task** | **EST** | **LST** | **EFT** | **LFT** | **Duration** | **Float** | | A | 0 | 3 | 4 | 7 | 4 | 3 | | B | 0 | 0 | 7 | 7 | 7 | 0 | | C | 0 | 1 | 9 | 10 | 9 | 1 | | D | 7 | 7 | 10 | 10 | 3 | 0 | | E | 10 | 10 | 12 | 12 | 2 | 0 | | F | 12 | 14 | 13 | 15 | 1 | 2 | | G | 12 | 12 | 15 | 15 | 3 | 0 | | H | 15 | 15 | 16 | 16 | 1 | 0 | | I | 16 | 16 | 18 | 18 | 2 | 0 | | | **Award 4 marks:**  0-1 mistakes  **Award 3 marks:**  2-3 mistakes  **Award 2 marks:**  4 mistakes  **Award 1 mark:**  5 mistakes  **Award 0 mark:**  > 6 mistakes |
|  | **Sub – total** | | **4** |
| d | Correct, highlighted critical path (B,D,E,G,H,I) | | 1 |
|  | **Sub – total** | | **1** |
| e | 18 | | 1 |
|  | **Sub – total** | | **1** |
| f | Dummy tasks are **artificial activities**, represented by a **dotted line** and **do not consume resources or require time**.  They are added into the network simply to **complete the logic**. | | Award 1 mark for complete description. |
|  | **Sub – total** | | **1** |
| g | C:\Users\s0132874c\Documents\Pages from oct10dippmreport-2.jpg | | Maximum 4 marks,  **SR:**  1 mark deducted for every major error or omission.  The Gantt chart should include:  • all tasks  • a clear scale  • a key  • a clear, correct, highlighted critical path  • correct durations, with dependencies (preferably with arrows) and float |
|  | **Sub – total** | | **4** |
|  | **Grand – total** | | **19** |
| 2a(i) | In bottom up testing the components at the lowest levels of the hierarchy are combined and tested first. The software is then put together by including successfully higher-level components.  a) Each individual module is tested as soon as it is written using pre-prepared test data. The data must include:  1. normal data which the procedure is designed to handle;  2. extreme values which test the behaviour of the module at the upper and lower limits of acceptability;  3. exceptional or invalid data which the procedure should reject rather than attempting to process it.  b) Each complete program in the system is tested. Data should be chosen which:  1. ensures that every route through the program is tested;  2. ensures that every statement in the program is executed at least once;  3. verifies the accuracy of the processing;  4. verifies that the program operates according to the original specifications.  c) usually carried out by developers to check that individual software modules work as specified by low level software designs.  d) tests that individual software components work correctly in conjunction with other units in the overall system. | | 1 mark for stating the selected strategy.  2 marks for description and elaboration of strategy. |
|  | **Sub – total** | | **3** |
| 2a(ii) | Top-down testing  The skeleton of the complete system is tested, with individual modules being replaced by 'stubs' which may, for example, display a message to say that a certain procedure has been executed. As individual modules are completed they are included in subsequent tests.  The coordinating module is written first. Then the modules at the next level in the structure chart are written, followed by the next level, and so on, until all of the modules in the system are done. Each module is tested as it is written. Because top-level modules contain many calls to subroutine modules, you may wonder how they can be tested if the low-level modules haven’t been written yet. The answer is stub testing. Stubs are two or three lines code written by a programmer to stand in for the missing modules. During testing, the coordinating module calls the stub instead of the subordinate module. The stub accepts control and then returns it to the coordinating module.  Checks that human operators can use the system effectively to carry out their tasks. The focus is on the interface design. | | 1 mark for stating the selected strategy.  2 marks for description and elaboration of strategy. |
|  | **Sub – total** | | **3** |
| b | Direct changeover  - new system replaces old system overnight; no transition time  - no need to run 2 systems side by side .·. less expensive  - immediate benefits from new system  - less disruptive  - more likely to work since it will have been fully tested first  Parallel implementation  - operate both systems together side by side  - duplication of work .·. more time consuming/more expensive  - good for training since both systems can be compared  - if new system fails have old manual system as a back up  Pilot implementation  - adopt new system at one of the garages  - can easily re-introduce old system if problems occur  - makes sure system fully works before adopting elsewhere  Phased implementation  - part of system (e.g. database) introduced initially for trials  - if it is OK, gradually introduce other parts of the new system  - if a problem occurs, can stop using it any stage  - allows training and staff to gain confidence in its operation | | 1 mark for chosen method  1 mark for the method, and  1 mark for illustration. |
|  | **Sub – total** | | **4** |
|  | **Grand – total** | | **10** |
| 3a | High 🡨 63  X = 0  High 🡨 Middle – 1 | | 1  1  1 |
|  | **Sub – total** | | **3** |
| 3b | 8 comparisons | | 1 |
|  | **Sub – total** | | **1** |
| 3c | e.g. in Python:  def BinarySearch(Low, High):  global Found  if Low>High:  return “Not found”  Middle=int((High+Low)/2)  if SearchData[Middle] == SearchItem:  Found = Middle  elif SearchData[Middle] < SearchItem:  BinarySearch(Middle + 1, High)  elif SearchData[Middle] > SearchItem:  BinarySearch(Low, Middle - 1) | | **Award:**  1  1  1  1 |
|  | **Sub – total** | | **4** |
|  | **Grand – total** | | **8** |
| 4a | * Data can only be accessed by the methods provided by the class * e.g. Name can only be accessed from the class Employee * Technical details of the methods and properties are hidden within the object. | | 1  1  1 |
|  | **Sub – total** | | **3** |
| 4b | * When one class is a subclass of another it can use its methods and attributes * e.g.Salesperson can use Getname() from Employee | | 1  1 |
|  | **Sub – total** | | **2** |
|  | **Grand – total** | | **5** |
| **5a** | A set of rules... To govern communication (between devices) | | 1 |
|  | **Sub – total** | | **1** |
| 5b | **Switch**   * A switch is a network device that acts as a common connecting point for various nodes or segments. * Switch breaks the network into LAN segments. * Switch filters frames * Switch forward messages between the computers attached to it   **Router**   * A computer specially designed **to switch packets**. * A router is a networking device that connects multiple networks. * Router is used to interconnect a LAN to WAN. * Router forwards packets from one **network** to another. It can have used to route data traffic between the LAN and the Internet. * A router is generally connected to several other routers. * If a router fails, traffic can be routed through other alternative path. | | Award 1 mark for any difference by comparing the switch and the router. |
|  | **Sub – total** | | **4** |
| 5c | |  |  | | --- | --- | | packet switching has no established route | circuit switching establishes a route along which to send packets | | packet switching means packets being sent on individual routes | circuit switching has packets all on same route | | packet switching message cannot be (easily) intercepted | circuit switching message can because all on same route | | packet switching packets need to be reordered | circuit switching packets remain in correct order | | packet switching maximises use of network | circuit switching ties up large areas of network | | | Award 1 mark per row chosen |
|  | **Sub – total** | | **3** |
| 5d | Answers may include:  Social:  • Less socialising because on computer all the time  • Sees other societies which will cause friction with…  • Company and other figures of authority  • Will raise expectations/wants  • Will increase knowledge of other societies…  • Give opportunity to learn about others/communicate directly  Any 2 points from the list.  Ethical:  • Use of other people’s work:  • Copyright…  • Plagiarism  • Use for educational purposes…  • • Use to spread understanding  1 mark for each point. Any 2 points from the list. | | 1 mark for each point. Any 2 points from the list.  **Guide:**  made a number of points about each of the social and ethical effects.  Included both problems and positive points.  Used appropriate technical terms throughout. |
|  | **Sub – total** | | **4** |
| 5e(i) | Any 2 from:   * Authentication techniques such as user names and passwords identify the user to the system * Without authentication anybody would be able to access data * Hackers would be able to amend/delete data without being prevented * Would have to rely on other methods such as firewalls to prevent unauthorised access. | | 1 mark for each point.  Any 2 points from the list. |
|  | **Sub – total** | | **2** |
| 5e(ii) | Access rights and password access:   * Confidentiality sensitive information is only released to authorized persons. * Integrity sensitive information content is always correct and is always delivered to the intended recipient * Availability To ensure that information can be made available to authorized persons who need. | | 1 mark for each point.  Any 2 points from the list |
|  | **Sub – total** | | **2** |
| 5e(iii) | A security policy describes the means that the organization will take to protect the confidentiality, availability, and integrity of sensitive data and resources, including the network infrastructure, physical and electronic data, applications, and the physical environment.  A security policy may include a consent to monitoring clause. The consent to monitor banner usually states that:   * Staff agree to be monitored in real time. * Staff are actually authorized to access or download specific data or files. * Staff and the network owner have a reasonable expectation of privacy. * Staff consent to reasonable law enforcement searches. | | 1 mark for each point.  Any 2 points from the list |
|  | **Sub – total** | | **2** |
| 5f | Intranet is an in-house website / client-server system / LAN (Local Area Network) which serves the people in an organisation.  It must be clear that the main purpose of an intranet is solely for internal use (e.g. sharing of information and resources) within an organisation and not accessible by the general public. | | 2  2 |
|  | **Sub – total** | | **4** |
| g | 1. Software-as-a-service (SaaS) - using a complete application running on someone else's system.  2. Platform-as-a-service (PaaS) - developing applications that use web-based tools, so they run in a software environment (i.e. a platform) provided by another company.  3. Infrastructure-as-a-Service (IaaS) - buying access to computing capacity over the internet, such as servers or storage.  1. Reduced maintenance, since hardware is managed by the service provider.  2. High availability, since cloud services can be available at any location.  3. Scalability, as only the storage volume and software required is paid for, and can be increase when needed.  4. Backup provision usually very comprehensive.  5. Reduced staff costs. | | **Services:**  1 mark each, total 3 marks.  Benefits:  1 mark each,  total 3 marks.  Any 3 points |
|  | **Sub – total** | | **6** |
|  | **Grand – total** | | **28** |
| **6a** | **Entities** | | |
|  | 4 correct entities  2-3 correct entities  0-1 correct entity | 2  1  0 | |
|  | **Relationships** | | |
|  | 3 correct relationships  1-2 correct relationships  0 correct relationship | 2  1  0 | |
|  | **Sub – total** | **4** | |
| 6b | **Attributes** |  | |
|  | All attributes identified for all 4 entities  All attributes identified for 3 entities  All attributes identified for 1-2 entities  Missing attributes for all 4 entities | 3  2  1  0 | |
|  | **Primary/Composite keys** |  | |
|  | Keys identified for all 4 entities  Keys identified for 3 entities  Keys identified for 1-2 entities  No correct keys for all entities | 3  2  1  0 | |
|  | **Sub – total** | **6** | |
| 6c | **Observations** |  | |
|  | Entity (Guest)  2 Data stores (Guest & Room files)  4 Processes | 1  1  4 (1 mark for 1 process) | |
|  | **Data Flow** |  | |
|  | All logical  ≥4 logical  < 4 logical | 2  1  0 | |
|  | **Sub – total** | **8** | |
| 6d(i) | matching CONDITION rules | 2  SR:  1 error deduct 1 mark  ≥2 errors 0 mark | |
|  | matching ACTION rules | 2  SR:  1 error deduct 1 mark  ≥2 errors 0 mark | |
|  | **Sub – total** | **4** | |
| 6d(ii) | Reductionfor guest N at column 5  Partial reduction | 2  1 | |
|  | **Sub – total** | **2** | |
|  | **Grand – total** | **24** | |
| 7(a) | |  |  | | --- | --- | | **ASCII** | **Unicode** | | 7 bits to represent data | 1 bytes to 4 bytes | | 96 printable characters represented are alphabet, digits, punctuation marks, some symbols  32 non-printable characters to control output | Millions of characters - all recognised languages in the world | | **Max 2 marks** (1 mark for each difference) | |
|  | **Sub – total** | **2** | |
| 7b | 20 + 21 + 24 + 25+ 28 + 210 + 212 + 215  = 38195 | M1 – correct working  A1 – correct answer | |
|  | **Sub – total** | **2** | |
| 7c | hex: 9516 3316 | M1 – correct working  A1 – correct answers | |
|  | **Sub – total** | **2** | |
|  | **Grand – total** | **6** | |
| 8a | A: print “ List is empty”, exit.  B: Start = temp.next  C:Previous = Current  D:Current = Current.next  E: Previous.next = Current.next | 1 mark for 1 correct statement | |
|  | **Sub – total** | **5** | |
| 8b(i) | **create new queue –**  Create (Q) | 1 | |
|  | **Sub – total** | **1** | |
| 8b(ii) | **add item to queue –**  Insert (Q, value, Length(Q)+1 ) | 1 mark for Insert method  1 mark for parameters | |
|  | **Sub – total** | **2** | |
| 8b(iii) | **delete item from queue –**  If IsEmptyList(Q) then exit - queue empty  Read (Q, 1)  Delete (Q, 1) | 1 mark for IsEmptyList, 1 mark for Read & Delete | |
|  | **Sub – total** | **2** | |
|  | **Grand – total** | **10** | |
| 9a |  | 2 marks for all correct  1 mark for 1 or 2 errors, 0 mark for > 2 errors | |
|  | **Sub – total** | **2** | |
| 9b | **A:** BinaryTree[TempPtr].Data <> SearchItem  **B:** BinaryTree[TempPtr].Data > SearchItem  **C:** TempPtr ← BinaryTree[TempPtr].RightPtr  **D:** RETURN TempPtr | 1 mark each | |
|  | **Sub – total** | **4** | |
| 9c | PROCEDURE PreOrderTraversal(Root)  OUTPUT BinaryTree[Root].Data  IF BinaryTree[Root].LeftPtr <> Null THEN  PreOrderTraversal(BinaryTree[Root].LeftPtr)  ENDIF  IF BinaryTree[Root].RightPtr <> Null THEN  PreOrderTraversal(BinaryTree[Root].RightPtr)  ENDIF  ENDPROCEDURE | 1 mark for parameter  1 mark for OUTPUT  1 mark for LeftPtr  1 mark for RightPtr | |
|  | **Sub – total** | **4** | |
|  | **Grand – total** | **10** | |