**Marking Scheme for 2019 JC2 Prelim Paper 1 (Lab-based)**

|  |  |
| --- | --- |
| **Evidence 1**  Program code for task 1.1 | ● open and close input file, correct mode |
| ● discard/skip first line of data |
| ● read every record into a data structure |
| ● organise record (dict or other data structure) |
| ● user input for start & end years |
| ● … with validation |
| ● … and error message(s) |
| ● method to total up visitors … |
| ● … for years specfied |
| ● Display record (with proper heading and format) |
| ● … for every year specified |
|  |  |
| **Evidence 2**  Table of 4 test cases Screenshots | ● Screenshot: valid start year and end year |
| ● Screenshot: invalid start year and/or end year |
| ● Screenshot: boundary start year or end year |
| ● Screenshot: start year later than end year |
|  |  |
| **Evidence 3**  Program code for task 2.1 and screenshot | ● declare function with parameter given |
| ● use correct index of 1 to (len-1) |
| ● ItemToBeInserted / CurrentItem assigned |
| ● correct condition for while loop |
| ● Array/CurrentItem correctly assigned |
| ● SomeList[CurrentItem + 1] ← ItemToBeInserted |
| ● return array |
|  |  |
| **Evidence 4**  Bubble sort function | ● declare function with parameter |
| ● for loops…compare each element with next |
| ● …swap elements |
| ● return array |
|  |  |
| **Evidence 5**  Amended program code to display number of comparisons Screenshot | ● declare counter |
| ● increment counter correctly |
| ● display counter correctly |
| ● screenshot |
|  |  |
| **Evidence 6**  Program code for Node class | ● declare Node class & show data types |
| ● …with private attributes data,priority,pointer |
| ● …get methods |
| ● …set methods |
| Program code for PQueue class | ● declare PQueue class & show data types |
| ● …private attributes Front, Rear |
| ● …nextfree set to 0 |
| ● …array ThisPQueue with 10 Nodes |
| ● …left pointers assigned correctly |
| Program code for JoinPQueue | ● Check for overflow with message |
| ● grab new node from free list and adjust pointers |
| ● assign Data and Priority to new node |
| ● insert empty queue…Front & Rear point to NextFree |
| ● insert as last node (lowest priority) |
| ● declare Previous and Current for traversing |
| ● …correct criteria for traverse queue |
| ● insert as first node… |
| ● insert as in-between node |
| ● …adjust pointers correctly |
| ● …adjust Front and Rear pointers |
| Program code for LeavePQueue | ● Check for underflow |
| ● assign Temp pointer to NextFree |
| ● assign NextFree to Front |
| ● assign Front to next node |
| ● assign nextFree node to Temp |
|  |  |
| **Evidence 7**  Program code for OutputPQueue Screenshot | ● Proper heading |
| ● Table in index order… |
| ● ...Data and Priority |
| ● …Front, Rear, NextFree |
|  |  |
| **Evidence 8**  Program code for task 3.3 Screenshot | ● Open file PATIENTS.txt for read and close |
| ● … every record into data structure |
| ● … split into Data and Priority |
| ● … Insert into priority queue using JoinPQueue |
| ● Display priority queue using OutputPQueue |
|  |  |
| **Evidence 9**  Program code for task 3.4 | ● Menu typed correctly |
| ● Option 1 call Pqueue |
| ● Option 2 call LeavePQueue |
| ● Option 3 call OutputPQueue |
|  |  |
| **Evidence 10**  Screenshots to test task 3.5 | ● Screenshot: remove patient |
| ● Screenshot: add ("Donny",2) |
|  |  |
| **Evidence 11**  Program code for displayboard Screenshot | ● declare **displayboard** with parameter |
| ● 2 for loops … |
| ● …display board using print correctly |
| ● copy and paste **puzzle1** into code |
| ● Screenshot of puzzle |
|  |  |
| **Evidence 12**  Program code for checkRow | ● declare **checkRow** with parameter |
| ● for every row… |
| ● …for every number in each row… |
| ● …check for each number |
| ● …and make a mark |
| ● return Boolean |
|  |  |
| **Evidence 13**  Program code for checkColumn | ● declare **checkColumn** with parameter |
| ● for every column… |
| ● …for every number in each column… |
| ● …check for each number |
| ● …and make a mark |
| ● return Boolean |
|  |  |
| **Evidence 14**  Program code for checkBlock | ● declare **checkBlock** with parameter(s) |
| ● …correct start coordinates for each block |
| ● …correct end coordinates for each block |
| ● …for every block |
| ● …for every number in each block… |
| ● …check for number |
| ● …and make a mark |
| ● return Boolean |
|  |  |
| **Evidence 15**  Program code for task 4.5 | ● call 3 functions |
| ● test puzzle1, puzzle2, puzzle3 |
| ● display puzzle before messages |
| ● suitable messages |
| ● screenshot |