Computer Science and Creative Technologies

Coursework or Assessment Specification

Module Details

Module Code	UFCFW4-30-2
Module Title	Design and Analysis of Data Structures & Algorithms
Module Leader	Dr Gordon Downie & Dr Elias Pimenidis
Module Tutors	Dr Kun Wei & Mr. James Lear
Year	2019-20
Component/Element Number	B / 2
Total number of assessments for this module	2
Weighting	60%
Element Description	Coursework B

Dates

Date issued to students	10 February 2019
Date to be returned to students	
Submission Date	19 March 2020
Submission Place	Blackboard - Online
Submission Time	14:00
Submission Notes	Please submit all files in a zipped folder labelled "DADSA1920B – Your Name"

Feedback

Feedback provision will be	Formative / Verbal, during practical sessions and during demonstration sessions Written on return of the marked work

Contents

Section 1: Overview of Assessment

This assignment assesses the following module learning outcomes:
Analyse requirements and select appropriate solutions
Design programmes that use appropriate data structures.
Implement data structures and the algorithms that maintain them, allowing for secure processing of the data.

The assignment is worth ***30%*** of the overall mark for the module.
Broadly speaking, the assignment requires you to select and implement appropriate data structures, design and implement algorithms and create the relevant software applications that will allow a user to store, update and manipulate the data relating to the operations of an animal sanctuary.
The assignment is described in more detail in section 2.
This is an individual assignment.
Working on this assignment will help you to develop the ability to relate organisational requirements into design choices for the storage and management of data. It will also help you practice your programming skills on Python and explore means by which the efficiency of your programming can improve.
If you have questions about this assignment, please ask your practical session tutor for clarifications.
Python libraries should not be used in creating code, neither should built-in data structure and algorithms templates should be utilised. The only readily available code you can reuse and customise is that for importing and reading a CSV file. **IN THE CASE OF A STUDENT USING LIBRARIES THE WORK WILL NOT BE ELIGIBLE TO ACCESS THE HIGHER MARK BANDS (MORE THAN 60%) FOR TASK 2 OF THE WORK.**

Section 2: Task Specification

Feeding young patients in a Pediatric Intensive Care Unit

The attached flowchart shows the decision making process for feeding patients in paediatric intensive care units (PICU) in a hospital.
Patients, depending on their condition, are classed as High Risk or Low Risk as shown in attached feeding decision making chart. Any other category is to be ignored. Patients are assessed by specialist nurses every few hours and their feeding maybe adjusted based on the various data that are recorded.
The hospital is looking to create a decision support system for the nurses to allow for speed, accuracy and consistency of the process. You are required to produce the core of the system that processes the data and allows the nurse to make an informed decision, quickly and consistently.
Data has been provided for you to complete the following tasks. There are 10 CSV files provided. Each file shows data for one patient over a feeding cycle of 5 days. **Please note that the data provided shows values in a realistic range, but is in no way related to a real person.**

You are required to complete the following tasks

Task 1

Given the attached flowchart, design suitable data structure(s) to employ and develop the algorithm to support this decision making process. Produce detailed pseudocode and / or relevant design diagrams - UML or any other equivalent.

Task 2

A. Save new data per patient as it is created in the cycle of evaluation indicated in the flowchart. Utilise the new data together with the data provided in each file to complete 5 day cycle of evaluation and feeding.

B. At the end of a 24 hour period, print on screen an update showing each patient's progress as they are fed and assessed through each day over a period of five days.

C. At the end of the 5 day cycle, rank the list of patients at the rate improvement they've shown and identify those who show lack of progress and they need to be referred to dieticians and specialist clinicians. At the top of a list you should show the patients that have been progressing their feeding regularly. These should be followed by patients that have had their feeding by one or more stoppages. Finally at the bottom of the list you should show the patients that have had to be evaluated by a dietician as their feeding had to be stopped for more than 8 hours.

Notes. A normal GRV should below 5ml x Patient weight in Kgs, or less than 250ml for children heavier than 40kgs.
q2h – stands for every 2 hours.

Task 3

A. Evaluate your design & implementation as to the effectiveness of the solution and the efficiency of the tools (data structures & algorithms) employed.

B. Discuss the concept of security of the data used here and the legal concerns. What should the hospital be doing to secure the data to be compliant with GDPR and other legislation in the UK?

Section 3: Deliverables

One folder in zip format (only) must be uploaded via the relevant link on the module's space on Blackboard. The link will be available two weeks before the due date and will be communicated to students via an email announcement.
The folder must contain:
All program files saved in Python format (3.6 or 3.7) – any other version will not be accepted and the work will not be marked resulting in a 0 (zero) mark for this coursework. (Tasks 2)
One word or PDF file containing the justification for the choice of data structures (Task 1)
One word or PDF document containing the Pseudo code and design diagrams (Task 1)
One word or PDF document containing your text for Task 3.
One text file with simple instructions of how to use / run your software

Section 4: Marking Criteria

The following table (please see next page) gives details of the marking criteria for this coursework.
Marks will be awarded for clear rationale justifying design choices.
Clarity in the pseudo code submitted allowing to map the full logic of the solution implemented is expected. A full set of diagrams showing interaction between

different parts of your system in processing the data must be submitted to allow you to access the top mark bands.
Code must be well structured, appropriately commented, neat and efficient. Clear use of functions and reduced repetitions of blocks of code are expected.
The use of GUI or other user interface will not attract any specific marks, but simplicity and efficiency of its design will be considered when awarding for an overall efficient system developed.

NOTE – No hard coded data will be allowed. Hard coded data in the submitted work will result in the work marked at 0 (zero).

Section 5: Feedback mechanisms

Formative / Verbal will be provided during practical sessions and during the demonstration sessions after submission of the work.
Written feedback will be provided on blackboard along with the return of the marked work.
Under normal circumstances the module team aims to return marked work on 26 April 2020.

Marking Criteria Table

	0-29	30-39	40-49	50-59	60-69	70-84	85-100	Mark & Advice for Improvement
Task 1	Choice of Data Structures appears random. Pseudo code lacks clarity / is incomplete.	There is vague attempt to justify choice of data structures. Pseudo code shows logic that does not address all requirements	Pseudo code addresses 51 to 60% of required features as specified in the requirements of Tasks 2 & 3	Pseudo code addresses 61 to 70% of required features as specified in the requirements of Tasks 2 & 3	Pseudo code addresses 71 to 80% of required features as specified in the requirements of Tasks 2 & 3	Pseudo code addresses over 80% of required features as specified in the requirements of Tasks 2 & 3. The design demonstrates elements of efficiency in the solution.	The design delivers all of the required features and goes beyond the requirements in such a way as to propose a solution that is fully efficient and will result in elegant program code.	
Task 2	Some of the subtasks are met successfully. The code is not neat and comments are sparse and unclear	50% of the expected functionality is delivered, but the successful implementation lacks evidence of intelligence and efficiency.	51 to 60% of the required functionality has been delivered, but issues with code structure, intelligence and efficiency persist	More sub tasks deliver the required functionality but this is no more than 70% of the expected. Code structure and comments leave room for quite a lot of improvement.	The functionality delivered has reached 80 % of the required level. The code appears more neat useful comments have been included in the code.	At least 90% of the code works efficiently and delivers the required functionality.	All tasks have been fully met. The code is elegant, well documented and efficient. The software exceeds the requirements offering a more complete and intelligent solution.	
Task 3	Text is vague and the justification of design choices is not clear. Reference to the GDPR and other legislation is not explicit and the impact is not clear.	The reference to data structures and algorithms is not clear as to why they are the right choice for the given problem. OR, the reference to legislation and assessment of the impact on the hospital is vague.	One of the two subtasks is clearly answered, while the answer for the other is still vague.	Both subtasks have been addressed, but the answers are brief and rather descriptive, not providing a clear explanation of the impact in each case.	At least one of the subtasks is addressed in detail and provides clarity as to the different options available and their impact in each case.	The answers for both the subtasks is addressed contain detail and provide good clarity as to the different options available and their impact in each case.	All subtasks have been addressed at an excellent level. The one on security legislation provides an answer of exemplary professional like analysis that could advise the hospital on future policy in	

							managing personal date	