



Complete, fold and seal
(see note 2 below)

Examination Answer Books

12 Pages

Instructions to Candidates

- Please give the information requested in the boxes below.
- Enter your name in the space provided to the right, fold over and seal down (using the label provided) to hide your name.
- The answers must be legibly written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
- Write your answers in ink.
- Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
- No part of this book is to be torn out; it must be handed in complete.
- If more books than one are used, fasten them together using the tags available from the invigilator(s).
- Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

SURNAME: LAYTON

FORENAME(S): MORGAN

WILLIAM OWEN

1	Student Number	199041
2	Title of Examination Paper	SOFTWARE DEVELOPMENT
3	Module Code	ALLA4017
4	Date and Time of Examination	
5	Examination Venue	
6	Seat Number	
7	Number of Answer Books handed in	

List the numbers of the questions you have attempted	Office Use
1a	2
1b	2
1c	5
1d	3+0
2a	10
2b	0
2c	8
3a	0
3b	0
4a	5
4b	4+1
4c	5
Total	45

✓
✓
✓
✓
✓
✓
✓
✓
✓
✓
✓
✓

195 ✓

471 ✓

- 1a. A queue data structure is a First in First out data structure. The implementation of a queue involves 2 pointers for example:

5	
4	
3	Jon
2	MORGAN
1	DAVE

Front pointer = 1
Back pointer = 3

An example of a queue data structure might be a printer queue where multiple machines in the same network are all requesting the printer to complete jobs. These requests would then be placed into the queue.

- b. A stack data structure are a Last in First out data structure. When a program that ~~adds~~ ^{stores} an address from a runtime it simply adds that address to the stack. Unlike a queue a stack only has 1 pointer. For example here I am adding a new name to my stack.

4
3 Jon
2 MORGAN
1 DAVE

4 TOM + New name is printer 4.
3 JON
2 MORGAN
1 DAVE

C. The term recursive algorithm means an algorithm or function when being defined is applied within its own definition. Quick sort is a recursive algorithm. It uses a theory of pivoting so the correct pivot position is found in the final sorted array. Items to the left of the pivot are smaller and items to the right are larger. Quick sort works best on large lists that are completely scrambled but has poor performance on small lists that are almost sorted. This is because it is recursive and calls on itself so will work through the entire list until it is sorted. This can be shown through big O notation as its best case at a completely scrambled list is $O(n \log(n))$ and worst case at a completely sorted is $O(n^2)$. For example a bubble sort would perform poorly at sorting a large list when compared with a quick sort function.

D. Object orientation has 4 main principles. Firstly is Encapsulation which is achieved when each object keeps its state private. Secondly we have abstraction which means that each object should only expose a high level mechanism so is essentially similar to encapsulation. Thirdly we have inheritance where we create a third class by deriving from another class which helps greatly with efficiency as it reuses fields and methods defined in the

parent class. Fourthly we have Polymorphism which is the provision of a single interface to entities of different types or the use of single symbol to represent multiple different types.

Please see C# 'Inheritance'!

4.

a) White box testing is where the internal structure of code is known to the tester. This is where the structure, branches, conditions, loops etc are tested to ensure they individually work rather than the complete functionality of the code.

Black box testing is where the tester has no knowledge of the code or program and is a method of testing the high level functionality of a program. Unlike white box testing very little knowledge of software development is required as the code does not have to be broken down like the white box method.

Grey box is essentially a combination of both forms of testing. This is where the individual components tested in white box testing are put together piece by piece to look for defects during integration that will hinder the performance of the program later on.

b) As units of the program are created and developed they are tested and integrated into a much larger working piece of code to make up a computer system. Units are sets of code made up of one or more modules.

As each part of the program is completed, it is tested to make sure that it works properly otherwise other units that interact with it could fail. This is called unit testing.

Unit testing is a component of test driven development or TDD. Unit testing only involves those ~~aspects~~ characteristics that are vital to the performance of the unit under test. This encourages the development of source code without immediate concerns about how such changes might affect the functioning of other units when these individual units have been tested and found to be efficient and error free groups of units can then be placed into components ready for integration testing.

C) Incremental development is where requirements are broken down into multiple standalone modules of software. The system is put into production when the first increment is complete. Often the "core" product after it has passed the design, coding and testing phases. One advantage is that it is ~~cost~~ cheap and efficient to execute however it is very rigid and does not handle changes in requirements very well.

Iterative development is where large ~~problem~~ projects are broken down and developed or enhanced. They are then put back together and tested as a whole rather than individually. This means feedback is available immediately which can be added to the next cycle. However it is inefficient as more communication between teams and customer are required which wastes time.