



Examination Answer Books

12 Pages

Instructions to Candidates

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right, fold over and seal down (using the label provided) to hide your name.
3. 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.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

Complete, fold and seal
(see note 2 below)

SURNAME: JONES

FORENAME(S): GEMMA

1	Student Number	0000380 CT196454
2	Title of Examination Paper	SOFTWARE DEVELOPMENT
3	Module Code	ACCA4017
4	Date and Time of Examination	9/1/19 5.30 - 7.10
5	Examination Venue	G42
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	1
1d	4+0
2a	8
2b	0+1
2c	10
3a	0
3b	0
4a	5
4b	4
4c	5
Total	42

195 ✓

441 ✓

1a) A queue is a data structure that goes 'First in first out' (FIFO). This means the first ~~one~~ item that is added to the queue, is the first one removed from it. There are front and back pointers, the front pointer points to the item that's been there the longest and the back pointer points to the newly added item.

An example of this would be a print queue. If you are printing several different documents, they are put into a queue so when they are printed the first document you clicked print on will be the first out of the printer. The queue is most appropriate for this as if they came out in a different order, it could mess up your document order.

2a) A stack data structure, also known as 'LIFO' - Last in first out, ~~is used when the~~ With a stack, there is always a stack pointer showing the top of the stack. An example of where this could be used is in Word or Excel in the 'undo' function. Text changes are put into a stack so if you wanted to undo anything the last item will be the first to be undone. This is beneficial as you can edit documents easily and can ~~go back~~ ^{undo} several steps in the LIFO order.

1c) Recursive algorithms simplify problems into sub problems of the same nature. The result of the first recursion is the input for the next recursion and so on.... An example of this would be the fibonacci sequence. Each number is an addition of the previous two:

e.g. 1 1 2 3 5 8 12 20 32

Quicksort would be used instead of another algorithm as it is faster and more efficient. Quick Sort also works in place on a stack so ~~it~~ uses a constant amount of memory.

1d) Object oriented programming focuses on objects and data rather than actions and logic. There are 3 main parts to object oriented programming. ✓
Encapsulation - This is when elements of your program are kept private. E.g. A program for public use, like a BMI calculator. The user does not need to see the code behind the program, this is why we encapsulate it and keep it hidden.

Polymorphism - This ~~is~~ ~~the~~ literally means 'many forms'. It's the ability to present the same object but under many different forms. For example, you can have a vehicle, there can be many different things that can also be called 'vehicles' but they are not the same, e.g. car, bus, motorcycle. They have properties that are similar. A numerical input can be classed as more than one data type. 236 could be an integer, float or a double. ✓

Inheritance. Similar to polymorphism to explain. Objects can be similar, share properties. You define parent and child classes. The child class inherits properties and methods from the parent class but they can also modify the behavior if required. ✓

- 2b) The code outputs as false because the Boolean value is false. "==" means 'equal to' and the code ~~8~~ is not equal to the answer. I have changed the operator to '!=' which means 'not equal to' and this now evaluates as 'true.'

3)

Input numbers

??



4a) Whitebox testing, also known as Clear box testing or open box testing is a testing method ~~to~~ where the tester/user knows the internal design/coding of the product/software. The tester is usually a coder.

Blackbox testing is when the internal design or code/structure is not known to the tester. Errors are found through missing functions. Eg someone could test a website - they do not know the coding behind it but can test it based on where the buttons pressed have the expected output.

Greybox testing is a combination of the both. The internal structure is partly known by the tester. It combines the input of developer and users which improves product quality.

4b) Unit test breaks code into pieces and tests each piece. C# does this as you're coding as it instantly displays errors in your code.

Unit testing is normally performed by software developers. ~~and~~

You write tests before the code

Scenario / Functional / unit Small tests.

Unit testing is performed using the white box testing method.

4C)

Incremental development is when software is designed / implemented and tested incrementally. Once one part is tested, the developers add more. An advantage of this is errors can be ironed out at each testing stage so software is developed quickly.

A disadvantage would be that each development phase is rigid and therefore do not overlap in any way.

In iterative development, a piece of software can be released in many versions. Each version will have bugs ironed out and elements added. An advantage of this is that an operational product is delivered / released with every iteration.

A disadvantage is that it can cost more to develop as more resources are needed to develop a functional product after every iteration.