



Examination Answer Books

12 Pages

Complete, fold and seal
(see note 2 below)

SURNAME:

Gallagher

FORENAME(S):

Ryan Dean

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.**

1	Student Number	142534
2	Title of Examination Paper	Application design & development
3	Module Code	M5X01025
4	Date and Time of Examination	14th May 2018 9:00
5	Examination Venue	Pembrokeshire College
6	Seat Number	4
7	Number of Answer Books handed in	1

List the numbers of the questions you have attempted	Office Use
1a → d	11
2a → c	8
3a → c	14
Office Use	33



i) An attribute is a declarative tag that is used to convey information to function about the behaviours of various elements like classes, methods, structures etc in your program.

more the values/data types

encapsulation is the practice of enclosing one or more items inside a physical/topical package.

this is especially done with keywords such as

"public" & "private" for access via predefined

"allow multiple = true"

methods preventing accidental change

ii) an object is a block of memory that follows the classes instructions.

a class is like a blueprint that holds information about each object.

a program class may have many objects, but an object can only be one class.

"Dog dog = new dog();" of

iii) inheritance is the ability to create a class

that inherits attributes & behaviours from an existing class / e.g. "Class Rectangle: Shape"

b) The House class initialises the dog object, whose only function is to bark. if would be a constructor for House

is called passing constructor

The House method House (Dog dog) to set the dog parameter object with the name "guard dog" so that it can be used in the Barkdog method.

each method calls upon another method in order to perform function properly.

The house method directly calls upon the change dog method, and the Barkdog method provided the dog object has been initialised) calls upon the Bark method.

2) a) What the house can contain 0 to many parts, this amount is calculated within the parts class.
House ~~interacts~~ holds the owner name & parts details. ✓

2) a) A house has one "owner".
The owner has a transient relationship with a car.
A house is comprised of 0 to many parts.
With parts referring to windows & doors

The house class receives the owner name & part details, and privately holds them.
~~Part~~ Part class will hold information about windows & doors.
The house gets the methods open(), and close() from parts class. (3)

2) b) Abstraction means to keep certain properties of a class unknown to the user, this means that classes will only inherit certain properties, and that only certain information is visible.

Abstraction is used here because details about - default task, and - current task are kept private. This means that the robot class only inherits perform task() from task.

Also the user will not be able to see details about the current task. everything is kept separate, one method for one function. (2)

Singleton pattern has been used here, as every method does one thing, and it does it well.

3) a) Single-responsibility principle says: a class should have only one reason to change. ✓
A class that follows this principle will have only one function, and will perform this function very well. This means that to update a system is very simple, as it will not bring up errors

as easily.
 it also means that debugging is drastically easier, and to detect a class will not cause a chain-reaction of bugs, because nothing else will depend on this class. ✓

⑥

3) c) Black box testing: testing where the inner workings of the system/code is NOT known by the tester.

White box testing: testing where the inner workings of the system/code ~~are~~ IS known by the tester.

White box testing is usually conducted by the individual(s) who developed the program, and black box is typically conducted by anybody else. ✓

It is very important to conduct both types of testing, as white box testing will look at things like variables, databases, code loops, whereas black box testing will look for functionalities, and it is very easy to overlook an error with only one method of testing. it is always best to test in as many ways as possible

⑤

3) b) A user can make an order, either a mobile phone app order, or a web order form.
 Separately from the process outlined above the user can display orders.

Also a user can confirm their order, which is immediately followed by taking a payment.

The "make order" function inherits from either "mobile app order" or "web order form".

display orders is a stand-alone function.

"confirm order" has a composition relationship with "Take payment", one cannot be an without the other.

The intended use of this application is of an online order page similar to eBay.

③

1) c) `Public void Cat(Selling the Cat Name)`

{
-name = name; ← done match.

`Public - owner (owner cat)`

no return parameter

Not a variable type

①

`Public void G (get cat)`

needs to be "+"

will throw error.

{
Console.WriteLine("Hello {0}", Cat.Name);

X

1) d) `House(dog-name);` ← sees guard-dog as "dog-name"

X - Guard Dog = dog-name

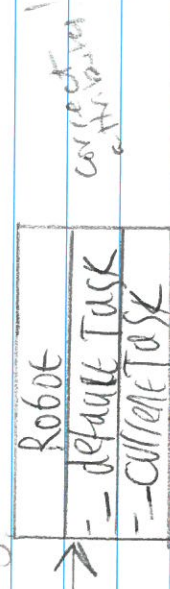
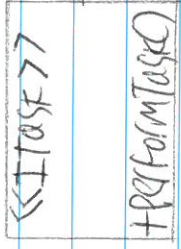
`Bang Door();`

②

`Dog dog-name = new dog(dog);`

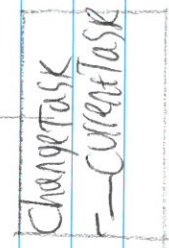
`dog-name.Bark();`

wrong direction



X

②



2) b) CONC. Using abstraction in OOP Programs means to give away all the important attributes of an object and keep only the visible ones. This is very important because say a rectangle & a square use same the same method to discover their perimeters "walkAround()" The square values will be wrong, because it was calculated as a rectangle. abstraction will ensure the square ~~objects~~ object ~~properties~~ the methods it needs, and overrides the rest. ✓ ①

