



**Complete, fold and seal
(see note 2 below)**

FORENAME(S): Thomas Gerwyn

12 Pages

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.**

List the numbers of the questions you have attempted	Office Use
1	13 1/2
2	10
3	10.5
Total	45%

$$\begin{array}{r} 34 \\ \hline 75 \end{array}$$

45% GT 1/6/18

1

a

- i Attributes represent variables inside of a class. For example if there were to be a 'person' class it could include such variables as 'name' 'gender' or 'Date of birth' as well as many others. 2½
- ii objects are used in programming using classes. Classes are used to represent objects and in turn that objects ~~attributes~~ characteristics are represented by the classes attributes. 2½
- iii Inheritance is used in programming when similar things will have common characteristics but also differ in other ways. For example all animals will have a species name but will then differ from that. For example a snake would not have legs or arms whereas a dog or cat would. In this case the animal class would contain 'species name' and then the snake and dog classes would inherit that and would then include that. But then include attributes like 'number of legs' or 'scree' and could include methods like 'Bark' or 'Hiss' 3

b

The code in appendix 1 represents a house and a dog. The dog contains a method to 'Bark'. In the house class a new dog object is added. The house class contains a method that when the door is knocked the Bark() method from the Dog class can be called to.

3

C

Public void Cat (String theCatName) - TheCatName is incorrect.
 it should instead be Name. Public should also be Private. X

Public ~~void~~ Owner (Owner cat) - Should be ~~private~~. X

Console.WriteLine ("Hello {0}", - Cat. - Name); - The - Cat. - Name is incorrect. Should instead be .Name, Name, name 1 1 2

Public Owner (Owner cat) - Should be void X

should include Console.ReadKey();

CatName() should include a Console.ReadLine to input a name X

D B

{

Method House

House New House X

~~House~~

House.BringDoor() ✓

Console.ReadKey();

}

2

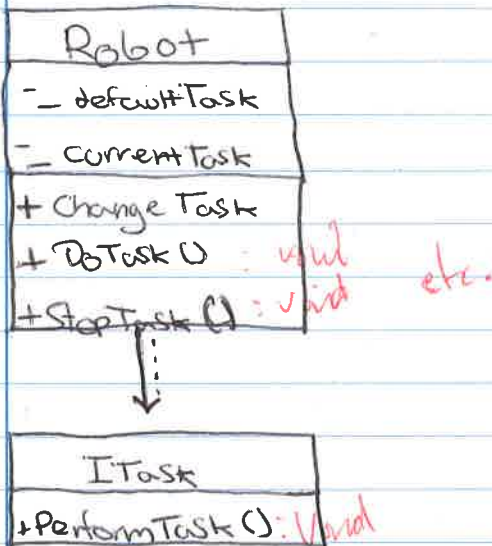
1. How the diagram represents a Diagram that shows a House class. The House class contains the main line and Owner class attached and parts class. The Owner class contains a Name attribute and a DriveCar method. Owners can have cars in order to drive them. The Parts class contains materials parts and then open and close methods. These are a Three methods can be used to open the window and Door interfaces. The car interface is attached to class 1 1 2

The owner class.

- B The code shown in appendix 4 is a robot class that completes task using the task interface. The Robot class can change tasks so that they can be completed. The class contains a DoTask() method which will complete ~~or~~ the current task and a StopTask() which will ~~contain~~ set the current task to the default ~~is~~ task. The code uses the factory pattern.

3 1/2

C



4

3

- a The Single responsibility Principle states that a class should have 1 responsibility and do only 1 thing. A class should have no reason to change. Making changes to classes should be avoided as the more that you change the more likely it is to go wrong. A class should include immutable objects (objects that are not changed once they are created). Using this principle will greatly improve the ~~quality~~ testability of the application. And with

reduce the likelihood of getting faults or bugs in code.

- b) The diagram in appendix B shows the features available in an e-commerce application from a user point of view. From the diagram we can see that the application is cross platform and allows the user to make an order from either a mobile application or a web order form. The application will then display the orders to the user. The user can then complete to order and a payment for the order can be taken. Once an order is made it should be passed to the display orders section before being sent to be confirmed.

- c) White and black box testing are 2 different forms of software testing. White box testing can be looked at testing the applications functions and making sure they work correctly for example checking a simple $2 + 2 = 4$ on a calculator. Simply making sure the application does as is intended. Black box testing is the form of testing in order to find the programs faults for example in a calculator using extremely large numbers trying to get the calculator to fail.