

- 1 OCR DogCare look after dogs while their owners are at work. They use a program developed in an object-oriented paradigm to store details about each dog.

Each dog is declared by using an instance of the class Dog. This has these private attributes:

- name
- breed
- height
- weight

The constructor method sets all the attributes to the values passed in as parameters.

Write pseudocode or program code to define the class Dog. You should include the attributes and constructor method.

You do not need to write the set or get methods.

This image shows a blank sheet of white paper with horizontal dashed lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no other markings or text on the page.

[5]

2(a) The Fibonacci sequence is a series of numbers. It starts with the number 0 and then 1. Each number after that is a sum of the two numbers before it.

The first seven numbers in the series are:

0
1
1 (i.e. 1 + 0)
2 (i.e. 1 + 1)
3 (i.e. 1 + 2)
5 (i.e. 2 + 3)
8 (i.e. 3 + 5)

Orla has written some code to show the first five numbers in the Fibonacci sequence (0,1,1,2,3) using the Little Man Computer (LMC) instruction set.

The LMC code that Orla has written contains an error.

START	LDA	MAX
	BRZ	END
	LDA	A
	OUT	
	ADD	B
	STA	B
	LDA	B
	STA	A
	LDA	MAX
	SUB	ONE
	STA	MAX
	BRA	START
END	HLT	
A	DAT	0
B	DAT	1
MAX	DAT	5
ONE	DAT	1

i. State the **five** outputs that Orla's code would give.

Output 1

Output 2

Output 3

Output 4

Output 5 [3]

ii. Orla has rewritten her LMC code to fix the error and added an additional DAT.

Complete the LMC code to output the first five correct numbers in the Fibonacci sequence (0,1,1,2,3).

START	LDA	MAX
	BRZ	END
	LDA	A
	OUT	
		
	STA	TEMP
		
	ADD	B
	STA	B
		
	STA	A
	LDA	MAX
	SUB	ONE
	STA	MAX
	BRA	START
END	HLT	
A	DAT	0
B	DAT	1
.....	DAT	0
MAX	DAT	5
ONE	DAT	1

[4]

(b) In Orla's LMC code, she used direct memory addressing.

Give **three** other modes of memory addressing.

- 1 [3]
- 2
- 3

(c) Since the development of high level languages, the use of assembly languages has reduced.

Give **two** reasons why in some circumstances programmers will choose to write code in assembly language.

- 1 [2]
- 2

3(a) A treasure game is being programmed using an object-oriented paradigm.

A class, Treasure, is used to store the treasure objects.

The design for the Treasure class, its attributes and methods is shown here.

class: Treasure
attributes: private value : integer private level : string
methods: new() function getValue() function getLevel()

- i. The constructor method takes a value as an integer, e.g. 100, and a level, e.g. "bronze", as parameters and assigns these to the attributes.

Write pseudocode or program code to declare the class Treasure.

You should define the attributes and constructor method in your answer.

You do **not** need to write the get methods.

[5]

ii. The get method `getLevel()` will return the appropriate attribute.

Write the method `getLevel()` using either pseudocode or program code.

[2]

iii. Describe the object-oriented programming technique being used in part 9(b)(ii).

[2]

(b) Describe **two** benefits of using an object-oriented paradigm rather than a procedural paradigm.

1 _____

2 _____

[4]

4 Describe **two** differences between assembly language and high-level languages.

Difference 1

Difference 2

[4]

- 5 A program is written using an object-oriented programming paradigm and uses a class called video to organise videos that are streamed to customers.

The class `video` has these attributes:

- name
- number of views
- star rating.

The constructor method will set the name attribute to the name that is passed in as a parameter. The constructor will also initially set the number of views to 0 and the star rating to 3.

- i. Write program code or pseudocode to declare the class `video` and initialise the required attributes as private.

You should include **both** the attribute definitions and the constructor method in your answer.

[7]

- ii. A public method called `updateviews()` will update the number of views after a video has been viewed.

This method is defined inside the `video` class.

Write program code or pseudocode for the method `updateviews()` to increase the number of views by one.

[2]

6 **Fig. 1** shows assembly code written using the Little Man Computer (LMC). The program calculates and outputs the total amount that is donated to a charity in any particular day.

Depending on the amount, an additional bonus may be added to each amount donated.

```
start      INP
           STA donation
           SUB hundred
           BRP bonus
nobonus     LDA total
           ADD donation
           STA total
           OUT
           BRA start
bonus       LDA total
           ADD donation
           ADD twenty
           STA total
           OUT
           BRA start
hundred     DAT 100
twenty      DAT 20
donation     DAT 0
total       DAT 0
```

Fig. 1

i. The program shown in **Fig. 1** is run **once** using **three** different inputs. Therefore, while the program is running once, it will output the updated total three times.

Give the total values that are output when the values **10**, **50** and **120** are input into this program.

Output for 10 -----

Output for 50 -----

Output for 120 -----

[3]

ii. Write LMC code that will reset the value of the memory location labelled `total` to zero and then stop the program.

[4]

iii. This program is run on a processor that allows pipelining.

Define the term 'pipelining'.

[3]

iv. Explain **one** benefit to a charity of using a processor that allows pipelining.

[2]

An object-oriented system is implemented to organize a company's information about staff attendance. Classes, objects, methods and attributes are used in this system.

- i. State the meaning of each of the following terms:

Object

.....

Method

.....

Attribute

.....

[3]

- ii. Each worker has a name and an attendance figure which can be between 0 and 100.

Write a definition for a fully encapsulated worker class, providing both get and set methods for all attributes. You do **not** have to write code for the constructor method.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[5]

8(a) * In assembly language, different modes of addressing memory can be used.

Discuss the different modes used. You should include:

- How the operand value is determined
- What an operand of 27 would refer to in that mode
- The reasons for requiring multiple modes of addressing

[12]

Handwriting practice lines consisting of 20 horizontal dashed lines.

(b) A program written using the Little Man Computer instruction set is shown in Fig. 1.

```

      INP
      STA numone
      INP
      STA numtwo
main  LDA numone
      SUB numtwo
      BRP pos
notpos LDA count
      OUT
      LDA numone
      OUT
      HLT
pos   STA numone
      LDA count
      ADD one
      STA count
      BRA main
numone DAT
numtwo DAT
one    DAT 1
count  DAT 0
```

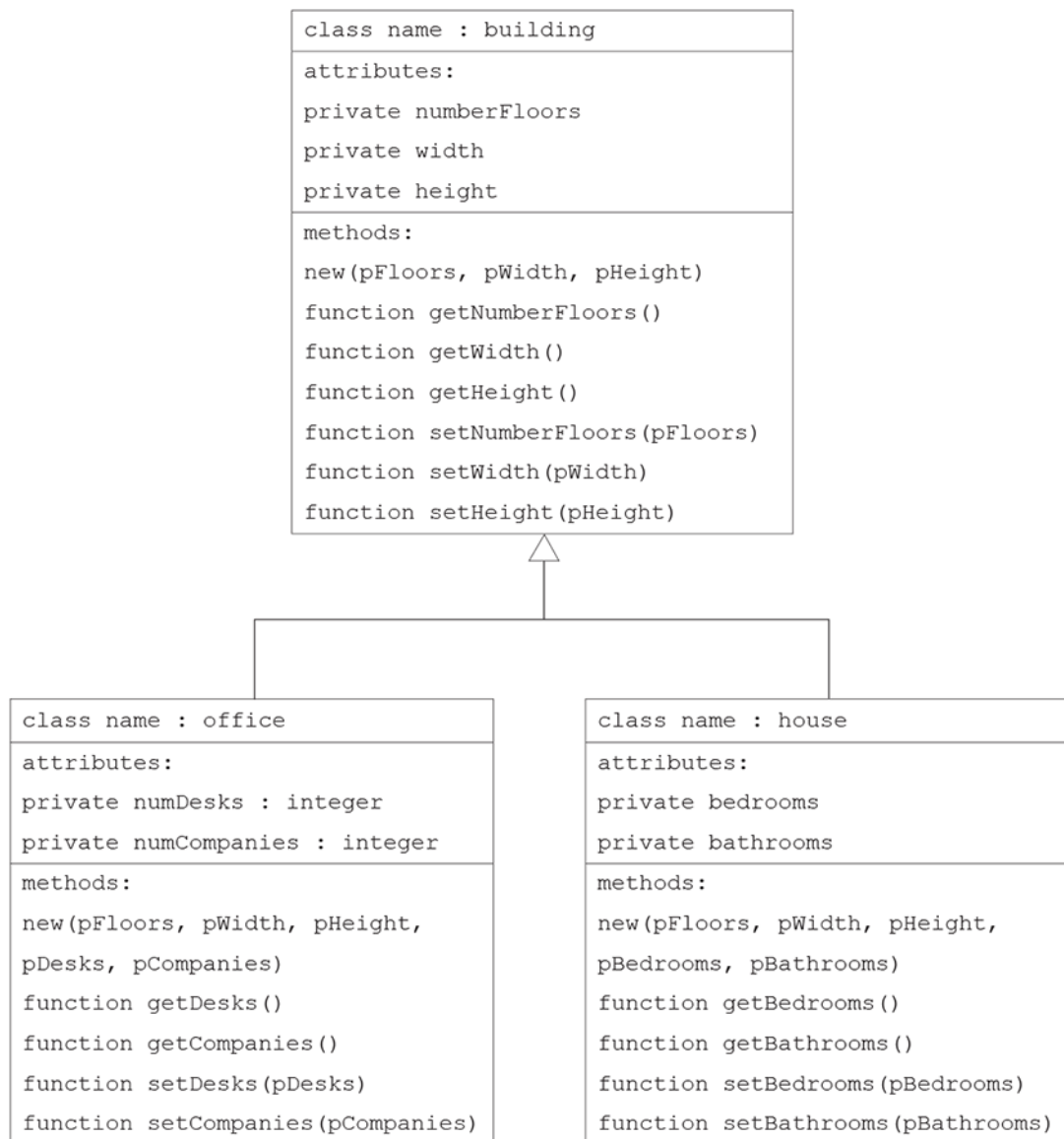
Fig. 1

Complete the table below to show the output(s) from this program given the inputs.

Inputs	Output(s)
12, 5	
18, 2	
16, 4	
3, 7	

[4]

9(a) Christoff is writing a program to simulate a city using object-oriented programming. He is designing classes to store different types of buildings and their location on the road. He has created the following plan for some of the buildings:



The classes office and house inherit from building.

Describe what is meant by inheritance with reference to these classes.

[2]

(b) Part of the declaration for the class building is shown.

Complete the pseudocode declaration by filling in the missing statements.

```
class building
  private numberFloors
  private width
  private .....
  public procedure new(pFloors, pWidth, pHeight)
    numberFloors = .....
    width = pWidth
    height = pHeight
  endprocedure
  public function getNumberFloors()
    return .....
  endfunction
  public function setNumberFloors(pFloors)
    //sets the value of numberFloors when the parameter is >= 1
    //returns true if numberFloors is successfully changed,
    //returns false otherwise
    if pFloors >= 1 then
      numberFloors = .....
      return true
    else
      return .....
    endif
  endfunction
endclass
```

[5]

- Define the attributes and constructor method in your answer. You do **not** need to write the get or set methods.

[6]

(d) Christoff develops a new class to store the houses in one road. His class design is shown:

class : houseRoad
attributes: private buildings(100) //array of class house private numberBuildings //records the number //of houses currently stored in the array //buildings
methods: new(building) function getBuilding(buildingNum) procedure newbuilding(pBuilding)

The method newbuilding() takes a new building as a parameter, and stores this in the next free space in the array buildings.

Write pseudocode or program code for the method newbuilding().

[4]

10(a) A supermarket uses an object-oriented approach to organise items that it offers for sale. Part of the class definition for the `ItemForSale` class is shown below.

```
class ItemForSale
    public itemName
    public price
    public discount

    ...

endclass
```

The `discount` attribute represents a percentage discount on the price. The discount can be between 0 and 50 (inclusive). All new items for sale initially have a discount value of 0.

- i. Write the constructor method for the `ItemForSale` class.

[4]

- ii. Write a line of code to create an object of type `ItemForSale` called `mushypeas` that has a name of “mushy peas” and a price of £0.89

[3]

iii. Write the `calculatePrice()` method, which applies the percentage discount to the price and returns the new value.

[3]

(b) The supermarket has previously had issues with discounts being set as values above 50.

Explain how encapsulation could be applied to the `ItemForSale` class to stop this problem from occurring.

You are **not** expected to write any code in your answer to this question.

[3]

(c) Some items in the supermarket are only available through home delivery. These items are the same as ItemsForSale with the following exceptions:

- the supermarket also stores the location of the stock
- the percentage discount allowed is up to 75 rather than the standard 50.

Explain how inheritance can be used to implement the above requirements.

[4]

11 * Procedural programming and object-oriented programming are two paradigms commonly used by programmers when developing computer games.

Discuss the advantages of using object-oriented programming over procedural programming when developing computer games. You should refer to inheritance, encapsulation and polymorphism in your answer.

Blank lined paper for writing.

END OF QUESTION PAPER

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
1	1 mark per bullet to max 5: • Suitable logic for class dog declaration • Suitable logic to define the 4 (private) attributes: name, breed, height, weight • Suitable logic to declare a <u>public</u> method for constructor (e.g. new or class name)... • taking <u>only</u> 4 different parameters in any order. • Suitable logic to set the values of each attribute MP4 - if it is obvious the candidate has used Pythonic syntax: allow the additional 5th parameter representing the object e.g. self, turtle <u>but this must be the first parameter listed.</u> MP2 - If they have done the above BP2 can be considered implicitly met (Python doesn't require attributes to be declared outside the constructor)	5	Mark in a vertical line against each MP. Ignore data types in attribute names Allow colon/empty brackets at the end of class def Example Solutions <u>Pseudocode style</u> <pre>class Dog private name private breed private height private weight public procedure new (nameIn, breedIn, heightIn, weightIn) name = nameIn breed = breedIn height = heightIn weight = weightIn endclass</pre> <u>Java / C# Style</u> <pre>Class Dog { private String name private String breed private float height private float weight public void Dog(String name, String breed, float height, float weight) { this.name = name this.breed = breed this.height = height this.weight = weight } }</pre> <u>Python Style</u> <pre>class Dog: def __init__(self, name, breed, height, weight): self.name = name self.breed = breed self.height = height self.weight = weight</pre> <u>Examiner's Comments</u>

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
					There were a range of responses to this question with the most common responses being written in pseudocode or python. There was a clear division between candidates who were comfortable writing Object Oriented Programming (OOP) code and class constructors and those that were not. Less successful candidates with little OOP experience often confused the class declaration with the constructor declaration. A few input the values rather than setting the attributes to the values passed in the parameters.
			Total	5	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
2	a	i	1 mark per bullet to max 3: • 0 • 1, 2, 4 • 8 <u>with no numbers after it</u>	3	CAO <u>Examiner's Comments</u> Most candidates were able to gain at least one mark for the first output on this question with many gaining full marks. A small minority of candidates simply copied the outputs given in the question stem.
		ii	1 mark for each correct line. <pre> START LDA MAX BRZ END LDA A OUT <u>LDA B</u> STA TEMP <u>LDA A</u> ADD B STA B <u>LDA TEMP</u> STA A LDA MAX SUB ONE STA MAX BRA STAR T END HLT A DAT 0 B DAT 1 <u>TEMP</u> DAT 0 MAX DAT 5 ONE DAT 1 </pre>	4	CAO Case for mnemonics can be ignored. Case for A, B, TEMP must be all caps. Penalise first error and allow FT. <u>Examiner's Comments</u> Most candidates were able to get the mark for the final answer gap and there was a good range of marks across the cohort.
	b		1 mark per bullet to max 3: • Immediate • Indirect • Indexed	3	DNA direct BOD 'index' for MP3 <u>Examiner's Comments</u> Most candidates were able to give indirect as a mode of memory addressing and there were many candidates who gained full marks on this question. A small number of candidates gave responses like by reference or value, and some gave direct which was mentioned in the question stem.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
	c		1 mark per bullet to max 2: • Quicker/more efficient to <u>translate</u> • Makes more efficient use of the CPU // memory // system resources // where a system may have limited resources • The programmer wants direct control over hardware/memory // to access machine specific functionality • Code might be written for a specific architecture • Compilers/interpreters may not be available	2	MP1 must be a comparison e.g. faster/quicker not just fast/quick <u>Examiner's Comments</u> More successful candidates were able to give two valid reasons. Less successful candidates tended to give responses like it is easier to read or easier for the computer to understand. Some candidates were confused between an assembler and compilers/interpreters. The most common correct responses were the programmer wanting direct control over the hardware and it needing to be written for specific hardware.
			Total	12	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
3	a	i	1 mark each - Defining class Treasure - Defining the private attributes value and level - Defining a new public procedure... ...Taking two parameters (integer and string) - Correctly assigning both parameters to the attributes e.g. <pre>class Treasure private value private level public procedure new(valueP, levelP) value = valueP level = levelP endprocedure endclass</pre>	5	Allow use of this/self or equivalent dependent on language <pre>public procedure new(value, level) this.value = value this.level = level endprocedure</pre> Python answers must either use comments to indicate private attributes or use the double underscore private attribute convention to be credited. <pre>self.__level self.level # private</pre> <u>Examiner's Comments</u> This question either tended to be very poorly answered or very well answered. There was a clear division between candidates who were comfortable writing OOP code and class constructors and those that were not. Weaker candidates with little OOP experience often confused the class declaration with the instantiation method or offered no response. Many candidates often assigned the parameters to the private attribute values rather than setting the attributes to the parameters being passed in. Where candidates gave responses in programming languages and the intent of the response was clear, marks were given. However, sometimes Python programmers did not make it clear (by convention/comment) that class attributes were private.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
		ii	1 mark each • get level method header with no parameter • Returning level attribute e.g. <pre>public function getLevel() return level endfunction</pre>	2	Note Python <i>self</i> will appear, but no other parameters <pre>def getLevel(self):</pre> <u>Examiner's Comments</u> While there were many good responses it was also noted that a significant number of candidates erroneously tried to send a parameter to a getter() function and then return the same parameter. Some candidates did not return a value or simply tried to print the value. Other errors included writing the getter method as a function call by omitting any indication that a function was being defined.
		iii	1 mark each • Encapsulation • Allowing an attribute to only be accessed/changed via a method	2	<u>Examiner's Comments</u> Some candidates randomly picked an erroneous OOP term like polymorphism. Those candidates who could correctly identify the use of encapsulation did not always go on to specifically state that this meant access was controlled by a public getter() method.
	b		1 mark each to max 4 e.g. • Code can easily be reused... • ...classes can be used in other programs • ...inheritance can be to extend upon existing classes • ...as a class can be based on an existing class • Easier to maintain.... • as classes can be modified or extended • ...debugging can be easier as encapsulation limits how attributes are changed. • Code can be more secure... • ... as access to attributes can be restricted to being via methods. • Better for coding as part of a team... • ...as classes can be distributed between team members.	4	1 mark per benefit identified and 1 mark per expansion. Max 2 benefits and 1 expansion per benefit. <u>Examiner's Comments</u> Many candidates were more successful on this part of Section B as it was an AO1 book learnt topic rather than the preceding AO2 OOP programming application elements. However, benefits were often poorly described, or OOP components were just named without a relevant expansion as to how they could be used.
			Total	13	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
4			• Assembly language uses mnemonics • HLL uses English-like words • Assembly language uses an assembler to convert to machine code • HLL uses a translator (compiler/interpreter) to convert to machine code • Assembly language is a one-to-one conversion to machine code • HLL may produce multiple lines of machine code per line of code // one-to-many • Assembly language requires more knowledge of the processor // allows direct control of the processor • HLL provides more abstraction // requires less knowledge of the processor • Assembly language is likely to be specific to the processor type used // is machine dependent • HLL is portable // can be used for multiple processor types // programmer can pick from a number of HLLs and paradigms // is machine independent	4	Mark in pairs. Allow examples (e.g. JMP, print) for MP1 and 2 Examiner's Comments There was a range of marks candidates could access for this question. The question asked for two differences and those candidates with a good understanding of language paradigms were able to gain full marks by giving the difference for both. Candidates tended to lose marks on this question by only giving one side of the difference, e.g. saying a high level language is translated using an interpreter or compiler but then not giving the other side that assembly language is translated using an assembler. Exemplar 2 Difference 1 ... assembly language is translated using an assembler into machine code while high level language use interpreters and compilers to translate Difference 2 ... assembly language use mnemonics to program while high level language use a language * closer to english * with syntax In this response the candidate has given two clear differences and has explained what the difference is for both.
			Total	4	

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
5	i - Class definition with identifier <u>video</u> - name, number of views and star rating attributes defined... ...As private - Constructor method definition <u>inside class definition</u>... ...that accepts only one parameter ...Name attribute set to parameter passed in - Views set to 0 and rating set to 3 either when initialised or in constructor.	7	Accept implementations in high-level languages (e.g. __ for private, class name used for constructor, no need for end of class definition in Python) BP1 - allow empty brackets. Do not allow anything in the brackets BP5 - ignore self if included as parameter <pre> class video private name private views private starrating public procedure new(newname) name = NewName views = 0 starrating = 3 end procedure end class </pre> Examiner's Comments This question was well answered by some candidates. The question asks for pseudocode or program code and candidates should be encouraged to do one or the other if given a choice, rather than a combination of the two. Many candidates did not use the information in the question stem to help them structure their answer and gave more than one parameter in the constructor. Exemplar 3 <pre> Class Video private name private number of views private star rating public procedure new(aName) name = aName number of views = 0 star rating = 3 end procedure end class </pre> This was a good clear example of an answer given in pseudocode. The candidate has declared the 3 given attributes as private, shown a constructor with one parameter and set name to the parameter and views and rating to 0 and 3. The candidate gained 7 marks.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
		ii	- Method definition <u>that is public</u> - View attribute incremented by one	2	public procedure updateviews() views = views + 1 end procedure View attribute must have the same name as part i <u>Examiner's Comments</u> Most candidates were able to gain at least 1 mark for this question. Those who were not given marks used pseudocode but did not state that the procedure was public, or they did not use the same attribute they had declared in Question 2 (g) (i).
			Total	9	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
6		i	• 10 • 60 • 200	3	1 mark per number Examiner's Comments This was generally well answered by candidates who had a good understanding of LMC. Candidates should be encouraged to trace through LMC programs with different values as well as writing them.
		ii	• Loads a value into the accumulator • Establishes a zero value (by use of DAT / SUB) • Stores a <u>zero value</u> into total • Program stops	4	Example 1 <pre>LDA zero STA total HLT zero DAT 0</pre> Example 2 <pre>LDA total SUB total STA total HLT</pre> BP1 can be given for any value being loaded into the accumulator e.g. INP If candidate writes LDA donation/total (case sensitive) they can get BP2 as they've used the labels from the question BP3 - total is case sensitive as given in the question BP4 - must not be given if the zero value will be attempted to be fetched e.g. HLT is placed after DAT Examiner's Comments This was generally answered well, and the majority of students were able to gain marks with most gaining 3 or 4 marks. Less successful responses over complicated the program leading to them making mistakes. A small number of candidates attempted to answer in pseudocode rather than LMC. Candidates should be encouraged to use the commands in Appendix 5d of the specification.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
		iii	- One instruction can be fetched while another is being decoded... ...and another is executed - The output of one <u>process/instruction</u> is the input of the next. - Concurrent processing of multiple instructions // completing multiple FDE cycles at once	3	For BP1, allow any 2 of the 3 parts of the FDE cycle For BP2, must give the other part of the FDE cycle not given in BP1 Do not award if explaining multiple cores working on different parts of FDE cycle Examiner's Comments Many candidates were able to gain at least 2 marks on this question. Some candidates were not awarded marks as they wrote about multiple cores or programs being fetched instead, of instructions. Exemplar 1 Pipelining is when a computer can fetch the next instruction whilst the previous is being decoded and the one before that is being executed. There is two types of pipelining: arithmetic and instruction. Pipelining allows multiple instructions to be processed at the same time. [3] The candidate has clearly described pipelining with correct terminology. They gained full marks for the description of one <u>instruction</u> being decoded while another is fetched and another is executed, as well as describing that it allows <u>multiple instructions</u> to be processed <u>at the same time</u>.
		iv	- More <u>instructions</u> can be carried out in a set amount of time // less time to execute the same number of <u>instructions</u> - Increasing the speed/performance/efficiency of the computer/program // quicker for the program to complete	2	Do not allow "each instruction is quicker to execute". BP2 has to be specific to the charity e.g. processing more donations Examiner's Comments Many candidates gained 1 mark for giving a benefit to the charity, but they did not go on to say why pipelining enabled that. Some candidates did not apply their answer to the charity, so were not awarded the mark for the benefit.
			Total	12	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
7		i	- Object – instantiated from class - Method – action object performs / link to procedure/functions - Attribute – value held by object / link to variable	3 AO1.1	
		ii	- Class definition statement - Defining name and attendance attributes - Appropriate get methods for name <u>and</u> attendance that return a value and have no parameter - Appropriate set methods for name <u>and</u> attendance that take a parameter ...that restricts attendance to be 0 to 100.	5 AO3.2	<u>Example answer</u> <pre> class Worker private name private attendance public function getName() return name end function public function getAttendance() return attendance end function public procedure setName(newName) name = newName end procedure public procedure setAttendance(newAttend) if newAttend >=0 and newAttend <=100 then attendance = newAttend end if end procedure end class </pre>
			Total	8	

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
8	a	Mark Band 3–High Level (9-12 marks) The candidate demonstrates a thorough knowledge and understanding of different modes of addressing. The material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well-considered. There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated. Mark Band 2-Mid Level (5-8 marks) The candidate demonstrates reasonable knowledge and understanding of different modes of addressing; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a sound discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence. Mark Band 1-Low Level (1-4 marks) The candidate demonstrates a basic knowledge of different modes of addressing; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided.	12 AO1.1 (2), AO1.2 (2), AO2.1 (3), AO3.3. (5)	AO1 Immediate addressing is where the operand holds the actual data to be used; no address referenced Direct addressing is where the operand holds the address that holds the data to be used. Indirect addressing is where the operand holds an address which is where the data to be used is stored Indexed addressing is where the operand holds an address which is offset using the Index Register to find the true address of the data to be used AO2 Immediate addressing; operand of 27 would refer to the value 27 Direct addressing; operand of 27 would tell the processor to fetch the data held at address 27. Indirect addressing; operand of 27 would tell the processor to fetch the data held at address 27, which itself would then be used an address to fetch data from. Indexed addressing; operand of 27 would be added to the value of the Index register and this would then be used as the address to fetch data from. AO3 Immediate addressing allows simple access to data with no fetch required, but limited by the data size of the operand. Direct addressing allows data to be fetched from memory. Data can potentially be larger in size that with immediate addressing but address range limited by size of operand. Indirect addressing allows a larger range of addresses to be accessed as address fetched. However, multiple fetches required to access data. Indexed addressing allows the Index register to be manipulated to access data stored sequentially e.g. in an array.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
			The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear. 0 marks No attempt to answer the question or response is not worthy of credit.		
	b		• 2, 2 • 9,0 • 4,0 • 0,3	4 AO2.2	Code finds integer division and remainder.
			Total	16	

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
9	a	1 mark per bullet up to a maximum of 2 marks, e.g: - When the child/derived/subclass class office/house takes on attributes/methods... ... from building / parent/base/superclass/ class	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	
	b	1 mark for each completed statement up to a maximum of 5 marks: <pre> private numberFloors private width private height public procedure new(pFloors, pWidth, pHeight) numberFloors = pFloors width = pWidth height = pHeight endprocedure public function getNumberFloors() return numberFloors endfunction public function setNumberFloors(pFloors) if pFloors >= 1 then numberFloors = pFloors return true else return false endif endfunction endclass </pre>	5 AO2.2 (3) AO3.2 (2)	Accept other specific language conventions that would correctly achieve the same outcomes.

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
	c	1 mark per bullet up to a maximum of 6 marks: • Class declaration for house with inherits building • Declaring bedrooms and bathrooms as private • New declaration ... • ... with all five parameters • Calling super constructor // equivalent with floors, width and height set • Setting bedrooms and bathrooms e.g. <pre>class house inherits building private bedrooms, bathrooms public procedure new(pFloors, pWidth, pHeight, pBedrooms, pBathrooms) super.new(pFloors, pWidth, pHeight) bedrooms = pBedrooms bathrooms = pBathrooms endprocedure endclass</pre>	6 AO2.1 (1) AO2.2 (2) AO3.2 (3)	
	d	1 mark per bullet up to a maximum of 4 marks: • Procedure header taking parameter • Adding parameter to array • ...at position numberBuildings • Incrementing numberBuilding e.g. <pre>procedure newbuilding(pBuilding) buildings[numberBuildings] = pBuilding numberBuildings = numberBuildings + 1 endprocedure</pre>	4 AO2.1 (1) AO3.2 (3)	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
	e		1 mark per bullet up to a maximum of 4 marks: • Creating a new instance of house with identifier houseOne • ...with the correct parameters • Creating a new instance of houseRoad named limeAvenue • ...sending houseOne as parameter to the constructor e.g. houseOne = new house(2, 8, 10, 3, 2) limeAvenue = new houseRoad(houseOne)	4 AO2.1 (1) AO3.2 (3)	
	f		To create an instance of an object from a class	1 AO2.1 (1)	
			Total	22	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
10	a	i	• Constructor method definition (.e.g new) • itemname, price passed in as parameters (must use different identifier names to the ones in the question) • ...and assigned to attributes • discount attribute assigned to 0.	4 AO3.2	<u>Example answer</u> <pre>public procedure new(nItemName, nPrice) itemname = nItemname price = nPrice discount = 0 endprocedure</pre> Look out for alternative notation e.g. this.itemname = itemName
		ii	• 1 mark for creating object with identifier mushypeas = • 1 mark for creating object as type ItemForSale • mark for parameters passed in as needed	3 AO3.2	<u>Example answers</u> <pre>mushypeas=new ItemForSale("mushy peas", 0.89)</pre> <pre>ItemForSale mushypeas = ItemForSale("mushy peas",0.89);</pre> <pre>mushypeas=ItemForSale(("mushy peas",0.89);</pre> Do not penalise for use of self parameter as used by languages such as Python. Must be correct case and spelling
		iii	• method definition for calculatePrice() • applies percentage discount • ...returns calculated value	3 AO3.2	Percentage discount must be applied correctly. <u>Example answer</u> <pre>function calculatePrice() newPrice = price - (price * discount/100) return newPrice end function</pre>
	b		• discount attribute made private • Set method created • ...that restricts value to maximum 50%	3 AO2.2	
	c		• Create new class • ...inheriting from / subclass of <u>ItemsForSale</u> • new attribute for stock location • calculatePrice() method overridden // new version of calculatePrice() implemented in subclass.	4 AO2.2	
			Total	17	

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
11	Mark Band 3–High Level (9-12 marks) The candidate demonstrates a thorough knowledge and understanding of procedural and object oriented programming. The material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well-considered. There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated. Mark Band 2-Mid Level (5-8 marks) The candidate demonstrates reasonable knowledge and understanding of procedural and object oriented programming; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a sound discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence. Mark Band 1-Low Level (1-4 marks) The candidate demonstrates a basic knowledge of procedural or object oriented programming; the material is basic and contains some inaccuracies. The	12 AO1.1 (2) AO1.2 (2) AO2.1 (23) AO3.3 (5)	AO1 Object oriented programming makes use of classes (templates) from which objects are made. Classes have attributes and methods Classes can be encapsulated by making attributes private and providing public access methods Object oriented programming supports inheritance which allows classes to use attributes and methods of parent classes. Object oriented programming supports polymorphism meaning that class attributes and methods can take on many different forms if required. AO2 OO approach would call on classes per type of player or enemy object Objects made from these classes, so one enemy class may generate many enemy objects, each with different values for their attributes (e.g. speed, energy) Special types of Player or Enemy objects could be instantiated from classes that inherit from the original Player/Enemy classes, but have attributes/methods of their own. Polymorphism would allow for the attributes/methods of Player/Enemy objects to behave differently from normal if required. AO3 OO promotes a modular approach (procedural through use of subroutines, OO through classes). OO is an abstraction of a real world problem, with classes for each type of things to be modelled and objects for each instance of these. OO has advantages in data security in that encapsulation forces developers/users to use methods (with their built in validation) to access/amend data stored in attributes. OO has advantages in efficiency of design where classes can be reused and can inherit from one another. Procedural programming struggles to support this. OO also offers flexibility through polymorphism.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
			candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided.		
			Total	12	