GCST

1.2.4 TYPES OF PROGRAMMING LANGUAGE

TOPIC WISE EXAM QUESTIONS

A-LEVEL OCR

GCSECOMPUTERSCIENCETUTOR.COM

1.2.4 Types of Programming Language	a) Need for and characteristics of a variety of programming paradigms. b) Procedural languages. c) Assembly language (including following and writing simple programs with the Little Man Computer instruction set). See appendix 5d. d) Modes of addressing memory (immediate, direct, indirect and indexed). e) Object-oriented languages (See appendix 5d for pseudocode style) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism.	Candidates need to understand that there are a variety of types of programming paradigms such as procedural, OOP, low-level, and that each has its strengths and weaknesses in specific scenarios, topics or areas. Candidates need to have knowledge and experience of using a procedural programming language for example Python, VB.NET etc. Candidates need to be experienced in using procedural programming features such as (but not limited to) variables, constants, selection, iteration, sequence, subroutines, string handling, file handling, Boolean and arithmetic operators. Candidates need to be able to read, trace, amend and write procedural program code. Candidates need to have an understanding of the purpose and need for assembly language. They need to be familiar with the instructions given in Appendix 5d. They should be able to read, write, trace and amend programs written in the Little Man Computer language. Candidates need an understanding of addressing, which should be integrated with assembly language. Candidates should have experience of using immediate, direct, indirect and indexed addressing in the writing, reading and tracing of programs written in assembly language. Candidates need to understand object-oriented code (as specified in the pseudocode guide). They need to have an understanding of classes, objects, attributes and methods. They need to understand the difference between private and public attributes and methods. Candidates need to understand encapsulation and the use of get and set methods to access private attributes. Candidates need to understand the purpose and principles of inheritance. Candidates need to have an understanding of polymorphism and how it can be used within a program. Candidates need to be able to read, trace, amend and write code that makes use of these object-oriented techniques.

4 A team of programmers create a robot that will be used in a factory. The robot will be able to do the work of multiple humans.

The programmers discuss whether to write the instructions for the robot in assembly language or a high-level language.

(a) Describe **two** differences between assembly language and high-level languages.

Difference 1 ....

....

....

....

Difference 2 ....

....

....

.... **[4]**

5 A `doCheck()` function takes an integer value as a parameter, carries out a series of calculations and returns an integer value.

The function is shown here.

```
function doCheck(number)
    temp = str(number)
    max = temp.length - 1
    total = 0
    for x = 0 to max
        total = total + int(temp.subString(x,1))
    next x
    return total MOD 10
endfunction
```

(a) State the value returned from the function when `doCheck(3178)` is called.

....

.... **[1]**

(b) Write an algorithm that will:

- allow the user to enter an integer value
- pass the value entered into the `doCheck()` function as a parameter
- store both the value input and the value returned from the function in a text file with name "storedvalues.txt"

You should write your algorithm using either pseudocode or program code.

[5]

1 A charity uses a desktop computer to record financial donations that it receives. The computer contains a single core, 2.4GHz processor with 2MB cache.

(c) **Fig. 1** shows assembly code written using the Little Man Computer (LMC). The program calculates and outputs the total amount that is donated to the 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]**

2 **(g)** A program is written using an object-oriented programming paradigm and uses a class called `video` to organise the videos that are streamed to customers.

[7]

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.

....

....

....

....

....

....

....

....

(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]**

5 A programmer creates this function shown in **Fig. 5** using a high-level language.

```
function mystery(x,y)
    total = x + y
    while x >= 10 then
        x = x - 10
        y = y - 10
        total = total + x + y
    endwhile
    return total
endfunction
```

Fig. 5

(a) **(i)** State the value output by the line `print(mystery(10,20))`

....

.... **[1]**

(ii) State the value output by the line `print(mystery(0,70))`

....

.... **[1]**

(iii) State the value output by the line `print(mystery(45,55))`

....

.... **[1]**

2021

6 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

(b) 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]

(c) Write an algorithm using pseudocode that has the same functionality as the code in **Fig. 1**. [4]

....

....

....

....

....

....

....

....

....

(d)* 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]**

....

....

....

....

....

....

....

....

....

....

....

....

....

....

7 A business uses an array with the identifier `wNames` to store workers' names. A variable with the identifier `top` is used to store the index of the last element to be added to the array, which is also the element which will next be removed.

`wNames`

0	1	2	3	4	5	6
Kirstie	Martyn	Louise	Alex	Anna		

`top`

4

(c) An object oriented system is implemented to organise further information about each worker's attendance. Classes, objects, methods and attributes are used in this system.

(i) State the meaning of each of the following terms:

Object ....

....

Method ....

....

Attribute ....

....

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

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

.... [5]

2 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
```

(a) 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]**

GCST

(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]**

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.

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

....

....

....

....

....

....

....

.... **[4]**

6 Two people play a counting game. The rules of the game are as follows:

- The first player starts at 1
- Each player may choose one, two or three numbers on their turn and the numbers must be in ascending order
- Players take it in turns to choose
- The player who chooses "15" loses the game.

For example, if the first player chooses three numbers (1, 2, 3) then the second player could choose one number (4), two numbers (4, 5) or three numbers (4, 5, 6). The first player then takes another go.

Write an algorithm using pseudocode that allows two players to play this game. The algorithm should:

- Alternate between player 1 and player 2
- Ask the player how many numbers they would like to choose, ensuring that this is between 1 and 3
- Display the numbers that the player has chosen
- Display a suitable message to say which player has won once the number 15 has been displayed. [8]

....

....

....

....

....

....

....

....

....

....

....

....

....

7 The table below shows the Little Man Computer instruction set.

Mnemonic	**Instruction**
`ADD`	Add
`SUB`	Subtract
`STA`	
`LDA`	Load
	Branch always
`BRZ`	
`BRP`	
`INP`	Input
`OUT`	Output
	End program

(a) Complete the table above to show the missing mnemonics and instructions. **[5]**

(b) Write a program using the Little Man Computer instruction set that will allow a user to input two numbers and then output the larger of the two numbers. The program should loop continuously. [6]

....

....

....

....

....

....

....

....

....

8* 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. [12]

3 A program written in the Little Man Computer instruction set is given below.

```
        INP
        STA    num
loop    LDA    total
        ADD    num
        STA    total
        LDA    count
        ADD    one
        STA    count
        SUB    num
        BRZ    end
        BRA    loop
end     LDA    total
        OUT
        HLT
one     DAT    1
num     DAT    0
count   DAT    0
total   DAT    0
```

(a) State what the program outputs are for the following inputs.

Input	Output
1	
2	
3	

[3]

(b) State what the purpose of the program is.

....

.... **[1]**

(e) The code uses direct addressing. Describe **one** other mode of addressing.

....

....

....

.... **[2]**

1 A digital coffee making machine has a CPU that uses the Little Man Computer Instruction Set.

(b) Part of the coffee making machine's code asks the user to press a button to select strength. The code outputs 1 which will switch on a green light to indicate a valid selection or outputs 0 to indicate an invalid selection.

The code is shown below:

```
              INP
              STA    entry
              LDA    max
              SUB    entry
              BRP    accept
              LDA    redLight
              BRA    printAndEnd
accept        LDA    greenLight
printAndEnd   OUT
              HLT
greenLight    DAT    1
redLight      DAT    0
max           DAT    5
entry         DAT
```

Fig. 1

(i) Tick the appropriate boxes below to indicate which inputs will result in a green light (i.e. code outputs 1) and which with a red light.

Input	Green Light	Red Light
1		
2		
3		
4		
5		
6		
7		
8		
9		

[2]

(iii) Write code in a high-level language or pseudocode that has the same functionality as the code in Fig. 1.

....

....

....

....

....

.... **[3]**

(iv)* Discuss the differences between assembly code and high-level languages. You should refer to:

- the advantages and disadvantages of writing programs in assembly code rather than a high-level language
- when each approach might be used
- why the coffee machine was programmed in assembly code.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[9]**

7 A taxi firm is investigating replacing its drivers with self-driving cars.

(b) The code for the self-driving system has been written using an object-oriented programming language.

It recognises obstacles in the road and then classifies them.
The class for `Obstacle` is shown below.

```
public class Obstacle
     private moving //Boolean value
     private distance //Real number given in metres
     private direction //Integer given as between 1 and 360 degrees

     public procedure new(givenMoving, givenDistance, givenDirection)
        moving=givenMoving
        distance=givenDistance
        direction=givenDirection
     endprocedure

     public procedure updateDistance(givenDistance)
            distance=givenDistance
     endprocedure

endclass
```

(i) Write a line of code to create an object called `bollard` of type `Obstacle` which is not moving and is 7.8 metres away in a direction of 8 degrees.

....

.... [2]

(ii) Describe an example of encapsulation in the class definition code above.

....

....

....

.... [2]

(iii) Describe the advantages of using encapsulation.

....

....

....

.... [2]

(c) The self-driving program recognises people as a special type of obstacle and the class `Person` should inherit the methods and attributes of `Obstacle`. People are treated like other obstacles except:

- when the `updateDistance` method is called, if the person is more than 2 metres away but is 5 metres (or less) away, the method `Controls.beepHorn()` is called.
- when the person is 2 metres away (or closer), the method `Controls.applyBrakes()` is called as well as `Controls.beepHorn()`.

Complete the class `Person`.

```
class Person .............................................................

      public procedure updateDistance(givenDistance)

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            ..............................................................................

            distance=givenDistance
      endprocedure
endclass
```

[5]

(d) Give **one** advantage and **one** disadvantage to the customers of the taxi using self-driving cars rather than drivers.

Advantage

....

....

Disadvantage

....

....

[2]

7 A meteorologist sets up a weather station to monitor temperatures throughout the year.

She classifies temperatures in one of four bands:

Band	Temperature Range (degrees Celsius)
Band A	10 or below
Band B	11–20
Band C	21–30
Band D	31 or above

The weather station records the temperature every day as an integer. At the end of the year the temperatures are stored in an array called `temperatures`.

Write a program in pseudocode that reads through this array and produces an output which shows the total number of days within each band. An example of such an output is shown below.

```
Band A: 93
Band B: 143
Band C: 98
Band D: 31
```

Ensure your code is efficient. [6]

9 **(c)** Write an assembly program (using the Little Man Computer instruction set) which repeatedly asks for a number until 3 is entered. When 3 is entered, the program should stop.

[3]

5 (a) Below is part of a program written using the Little Man Computer instruction set. This section of code can exit by either jumping to the code labelled `pass` or `fail` depending on what value is in the accumulator when the code is run.

```
test    SUB    ten
        BRZ    pass
        BRP    test
        BRA    fail

ten     DAT    10
```

(i) Explain what the line `ten DAT 10` does.

....

....

....

....

....

.... **[3]**

(ii) Complete the table below determining whether the program branches to `pass` or `fail` given the following values in the Accumulator when it is run.

Starting value in Accumulator	`pass` or `fail`
29	
30	
31	

[3]

(b) The complete program is shown below:

```
        INP
main    STA    entry
        BRA    test
fail    LDA    entry
        ADD    one
        BRA    main

test    SUB    ten
        BRZ    pass
        BRP    test
        BRA    fail

pass    LDA    entry
        OUT
        HLT

entry   DAT
ten     DAT    10
one     DAT    1
```

(i) Give **one** instruction in the program that when executed, changes the value in the Accumulator.

....

.... **[1]**

(ii) Give **one** instruction in the program that when executed, changes the value in the Program Counter.

....

.... **[1]**

(iii) State the value the code outputs for the input 18.

....

.... **[1]**

(iv) State the value the code outputs for the input 37.

....

.... **[1]**

(v) Describe the purpose of the program.

....

....

....

.... **[2]**

4 A delivery company sends parcels across the UK.

(a) The company charges on the following basis:

- Parcels that have a volume of less than 0.3 m^3 and weigh less than 4 kg cost £5 to send.
- All other parcels cost £20 per m^3 or £2 per kg, whichever is greater.

Examples

Parcel A weighs 2.5 kg, has a volume of 0.1 m^3 and costs £5 to send.
Parcel B weighs 6 kg, has a volume of 0.2 m^3 and costs £12 to send.
Parcel C weighs 6 kg, has a volume of 0.8 m^3 and costs £16 to send.

The function `getCost` takes in the `volume` and `weight` of a parcel and returns the cost.

`getCost(2.5,0.1)` returns `5`
`getCost(6,0.2)` returns `12`
`getCost(6,0.8)` returns `16`

Complete the pseudo-code below so that the function `getCost` returns the correct cost.

```
function getCost(weight, volume)

    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................

endfunction
```

[4]

If you found this useful, drop a follow to help me out!

THANK YOU!

GCST