GCST

2.2.1 PROGRAMMING TECHNIQUES

TOPIC WISE EXAM QUESTIONS

A-LEVEL OCR

GCST

SPEC

2.2.1 Programming techniques

a) Programming constructs: sequence, iteration, branching.

b) Recursion, how it can be used and compares to an iterative approach.

c) Global and local variables.

d) Modularity, functions and procedures, parameter passing by value and by reference.

e) Use of an IDE to develop/ debug a program.

f) Use of object oriented techniques

Candidates need to understand the constructs of sequence, iteration and branching. They must be able to use these constructs independently of each other, and combine them to produce a solution. These include the selection statements of IF (including ELSEIF and ELSE) and select case statements. These also include both condition based iteration (e.g. while, repeat until) and count controlled iteration (e.g. FOR) – as well as how condition based iteration can be used as count controlled.

Candidates need to have an understanding of the principle of recursion and the key features that produce a recursive algorithm such as a stopping condition. They need to be able to read and trace recursive functions, write recursive functions, and translate a recursive function to an iterative solution and vice-versa. Candidates need to have an understanding of the benefits and drawbacks of using both a recursive and iterative solution.

Candidates need to be able to read, create and trace code (for example using a trace table) that use these three constructs.

Candidates need to understand the use and need for variables in a program and must understand the difference, benefits and drawbacks of both global and local variables.

Candidates must be able to recognise where local and global variables are used and the impact that these have on a program, for example the amount of memory used. Candidates need to understand how a program using global variables can be changed to use local variables – and vice-versa.

Candidates need to understand what is meant by modular code, and how this can be produced using functions and procedures. Candidates need to understand the differences between functions and procedures and how each is used within a program. Candidates need to be able to read, trace and write code that uses functions and procedures. Candidates need to understand the purpose and use of parameters within a program, and how they are used in functions and procedures. Candidates need to be able to read, trace and write code that makes use of parameters.

Candidates need to understand the difference between passing a parameter by value and by reference, and they need to understand the benefits and drawbacks of each – recommending which should be used for a given situation. Candidates need to be able to read, trace and write code that makes use of parameters passed both by value and by reference.

Candidates should have had experience of using an IDE to produce code. Candidates need to understand how an IDE can be used to produce code, and understand the range of features and tools that are within an IDE, that can be used to help produce and debug a program.

Candidates need to understand the purpose of object-oriented code. They need to have an understanding of classes, objects, properties, attributes, methods. They need to understand the difference between private and public properties, attributes and methods. Candidates need to understand encapsulation and the use of get and set methods to access private properties. Candidates need to understand the purpose and principles of inheritance, super-classes, parent-classes and sub-classes. Candidates need to have an understanding of polymorphism and how it can be applied to classes. Candidates need to be able to read, trace and write code that makes use of these object-oriented techniques. Candidates need to be able to interpret class diagrams to produce class definitions. Candidates need to be able to identify where object-oriented programming can be used in a solution, and derive an object-oriented solution for a given scenario.

4 A programmer has designed a program that includes a reusable program component.

(a) The reusable program component is a function called `isInteger()`. This will take a string as an argument and then check that each digit is between 0 and 9. For example if 103 is input, it will check that the digits 1, 0 and 3 are each between 0 and 9.

The `asc()` function returns the ASCII value of each digit. For example `asc("1")` returns 49.

The ASCII value for 0 is 48. The ASCII value for 9 is 57.

```
01   function isInteger(number)
02     result = true
03     for count = 0 to number.length-1
04       asciiValue = asc(number.substring(count, 1))
05       if not(asciiValue >= 48 and asciiValue <= 57) then
06         result = false
07       endif
08     next count
09     return result
10   endfunction
```

(i) Identify **one** identifier used in the function `isInteger()`.

.....[1]

(ii) Give the line number where the branching (selection) construct starts in the function `isInteger()`.

.....[1]

(iii) Give the line number where the iteration construct starts in the function `isInteger()`.

.....[1]

(b) Describe the purpose of the following lines in the function `isInteger()`.

Line 03 .....

.....

Line 04 .....

.....

Line 09 .....

.....

[3]

5 A recursive pseudocode function, `recursiveAlgorithm()`, is shown.

```
01  function recursiveAlgorithm(value)
02    if value <= 0 then
03      return 1
04    elseif value MOD 2 = 0 then
05      return value + recursiveAlgorithm(value - 3)
06    else
07      return value + recursiveAlgorithm(value - 1)
08    endif
09  endfunction
```

(a) Describe the key features of a recursive algorithm.

You may refer to the function, `recursiveAlgorithm()` in your answer.

.....

.....

.....

.....

.....

..... **[3]**

(b) Trace the recursive function, `recursiveAlgorithm()`, and give the final return value when called with `recursiveAlgorithm(10)`. You may choose to use the table below to give your answer.

.....

.....

.....

.....

.....

.....

.....

Function call	value	return

Final return value ..... **[5]**

6 Octal is a base 8 number system.

To convert a denary number to base 8:
- the denary value is divided by 8 and the remainder is stored
- the integer value after division is divided by 8 repeatedly until 0 is reached
- the remainders are then displayed in reverse order.

Example 1:
Denary 38

38 / 8 = 4 remainder 6	6
4 / 8 = 0 remainder 4	4

Octal = 46

Example 2:
Denary 57

57 / 8 = 7 remainder 1	1
7 / 8 = 0 remainder 7	7

Octal = 71

Write an algorithm to:
- take a denary value as input from the user
- convert the number to octal
- output the octal value.

You do **not** need to validate the input from the user.

Write your algorithm using pseudocode or program code.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

7 **(d)** The programmer uses an Integrated Development Environment (IDE).

Complete the table by identifying **and** describing **three** IDE features that can help the programmer to develop, or debug a program.

IDE feature	Description

[6]

9 A text-based computer game allows a user to dig for treasure on an island. The island is designed as a grid with 10 rows and 20 columns to store the treasure. Each square is given an x and y coordinate. Some of the squares in the grid store the name of a treasure object. Each treasure object has a value, e.g. 100 and a level, e.g. "Bronze."

(b) The 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]**

(c) A class, `Board`, is used to store the 10 row (x coordinate) by 20 column (y coordinate) grid.

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

class: Board
attributes: private grid : Array of Treasure
methods: new() function getGridItem(x, y) function setGridItem(x, y, treasureToInsert)

The constructor initialises each space in the grid to a treasure object with `value` as -1 and `level` as an empty string.

Complete the following pseudocode for the constructor method.

```
public procedure new()
  for row = ..... to 9
    for column = 0 to .....
      ..... [row, column] = new Treasure(.....,"")
    next .....
  next row
endprocedure
```

[5]

(d) A procedure, `guessGrid()`:

- takes a `Board` object as a parameter
- accepts the row (x) and column (y) coordinates from the user
- outputs "No treasure" if there is no treasure found at the coordinate (level is an empty string)
- if there is treasure at that coordinate, it outputs the level and the value of the treasure in an appropriate message.

Write the procedure `guessGrid()` using either pseudocode or program code.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[7]**

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

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

(f)* The main program initialises a new instance of `Board`. The programmer is considering declaring this as a global variable or as a local variable and then passing this into the subroutines that control the game.

Compare the use of variables and parameters in this game.

You should include the following in your answer:

- what is meant by a local variable and global variable
- how local and global variables can be used in this program
- the use of passing parameters by value and by reference.

[9]

3* Nina is writing a computer game using an Integrated Development Environment (IDE). Her friend, James, is writing a computer game using a text-editor which will allow James to create and edit text. James will use a separate compiler.

Discuss the differences between writing and debugging a program using an IDE and a text-editor.

You should include the following in your answer:

- features that are used when writing code
- features that are used when debugging code
- the benefits of using an IDE instead of a text-editor. [9]

4 A function, `toBinary()`, is needed to calculate the binary value of a denary integer between 0 and 255.

`toBinary()` needs to:

- take an integer value as a parameter
- divide the number by 2 repeatedly, storing a 1 if it has a remainder and a 0 if it doesn't
- combine the remainder values (first to last running right to left) to create the binary number
- return the binary number.

For example, to convert 25 to a binary number the steps are as follows:

25 / 2 = 12 remainder 1
12 / 2 = 6 remainder 0
6 / 2 = 3 remainder 0
3 / 2 = 1 remainder 1
1 / 2 = 0 remainder 1

return value = 11001

(a) Write the function `toBinary()`.

You should write your function using pseudocode or program code.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

(b) The main program:

- asks the user to enter a denary number between 1 and 255
- checks that the input is valid between 1 and 255
- If valid call the function `toBinary()` and pass the input as a parameter
- outputs the return value
- If not valid, repeatedly asks the user to input a number until the number is valid.

Write the algorithm for the main program.

You should write your algorithm using pseudocode or program code.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

5 Layla writes a pseudocode algorithm to:

- input 20 positive numbers into a 0-indexed 1-dimensional array
- output the average (mean) number as a decimal
- output the smallest number
- output the largest number.

The pseudocode algorithm is shown. It contains various errors.

```
01 total = 1
02 smallest = 9999
03 largest = -1
04 for x = 0 to 21
05   dataArray[x] = input("Enter a number")
06   total = total + dataArray[x]
07   if dataArray[x] < largest then
08     largest = dataArray[x]
09   endif
10   if dataArray[x] < smallest then
11     smallest = dataArray[x]
12   endif
13 next x
14 print("Average = " + total * 20)
15 print("Smallest = " + smallest)
16 print("Largest = " + largest)
```

(a) (i) Identify the construct used on lines 01 to 03 in the algorithm.

.....

..... **[1]**

(ii) Identify the construct used on lines 10 to 12 in the algorithm.

.....

..... **[1]**

(d) `dataArray` is defined as a local variable within the main program.

(i) State what is meant by a 'local variable'.

.....

..... **[1]**

(ii) Give **one** benefit and **one** drawback of declaring `dataArray` as a local variable in the main program.

Benefit .....

.....

Drawback .....

.....

[2]

GCST

2022

1 A computer program stores data in an array named `words`.

(b) The array `words` is defined as a global variable and contains these values:

"house"	"boat"	"car"	"telephone"	"garden"	"spice"	"elephant"

The pseudocode function `useWords()` here uses the global array `words`.

The number of words in the array `words` is passed as a parameter.

```
function useWords(numberOfWords : byVal)
  contents = ""
  for count = 0 to numberOfWords - 1
    contents = contents + words[count] + " "
  next count
  return contents
endfunction
```

(i) Identify **two** variables in the function `useWords()`.

1 .....

2 .....

[2]

(ii) `numberOfWords` is a parameter passed by value.

Describe the difference between passing a parameter by value and by reference.

.....

.....

.....

..... [2]

(iii) Rewrite the function `useWords()` to use a while loop instead of a for loop.

The function header and close have been written for you.

Write your answer using pseudocode or program code.

```
function useWords(numberOfWords : byVal)
```

.....

.....

.....

.....

.....

.....

.....

.....

```
endfunction
```

[4]

(c) Give **one** benefit and **one** drawback of declaring the array as a global variable instead of a local variable.

Benefit .....

.....

Drawback .....

.....

[2]

(d) Describe **one** feature of an Integrated Development Environment (IDE) that can be used to help write the program **and one** feature that can be used to help test the program.

Write .....

.....

.....

.....

Test .....

.....

.....

.....

[4]

(e) Functions and procedures are reusable components.

Give **two** benefits of writing a program with reusable components.

1 .....

.....

2 .....

.....

[2]

7 A program uses the recursive function `calculate()`. The function is written in pseudocode.

```
1.  function calculate(number : byVal)
2.      if number == 1 then
3.          return number
4.      else
5.           return number + calculate (number - 1)
6.      endif
7.  endfunction
```

(a) (i) Give the line number in the algorithm `calculate()` where a recursive call is made.

..... **[1]**

(ii) State **two** features of any recursive algorithm.

Feature 1 .....

.....

Feature 2 .....

.....

[2]

(b) Trace the recursive function `calculate()` and give the final return value, when the following function call is run:

```
calculate(5)
```

You may choose to use the table below to give your answer.

.....

.....

.....

.....

Function call	number	return
calculate(5)		

[5]

(c) Give the pseudocode function call that would return 55 from the recursive function `calculate()`.

.....

..... **[1]**

1 Ruhail owns ten different function rooms which can be hired by different business customers to hold meetings. He would like a program to manage the booking process of each room.

Customers should be able to enter the date they want to hire a function room, and then a list of available rooms will be displayed. Customers can then select which room they want to hire. Customers can then enter their payment details which are then checked and then a confirmation email is sent to the customer.

(b) Ruhail will make use of an Integrated Development Environment (IDE).

State the purpose of an IDE.

.....

..... **[1]**

(c) State **two** different programming constructs and give an example of how Ruhail could use each construct when creating his program code.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

2 Logan is writing a program for his customers to be able to buy his gym equipment. In the program, once a customer has selected the items they want to buy, a procedure, `checkDetails`, will be called. This procedure will check that the customer has input their telephone number and also check that it is at least 11 characters long.

(iii) When setting up the additional procedures in his program, Logan will use a mixture of parameter passing by reference and by value.

State the difference between parameter passing by reference and parameter passing by value.

.....

.....

.....

..... **[2]**

4 Given the following procedure:

```
01 procedure generate(number)
02       a = 0
03       while number > 0
04             if number MOD 2 == 0 then
05                   a = a + 2
06                   print(a)
07                   number = number - 2
08             else
09                   a = a + 1
10                   print(a)
11                   number = number - 1
12             endif
13       endwhile
14 endprocedure
```

(a) Explain why = is used on line 11 of the procedure `generate` instead of ==.

.....

.....

.....

..... **[2]**

(b) State the values printed by the procedure `generate` when `number` = 8.

.....

..... **[1]**

(c) State the values printed by the procedure `generate` when `number` = 7.

.....

..... **[1]**

(d) Describe the purpose of the `MOD` operator on line `04` of the procedure `generate`.

.....

.....

.....

..... **[2]**

6 Kylie buys used games consoles and then sells them to make a profit. She sells her products in multiples of £5 such as £30, £55 and £95. Kylie only accepts £50, £20, £10 and £5 notes from her customers.

Kylie has written an algorithm which will calculate the amount of change needed by stating how many £20, £10 and £5 notes are needed.

The program should output the minimum number of notes required. For example if £35 change is required then it should output 1 x £20 and 1 x £10 and 1 x £5.

```
01 total = input("Enter total price of goods")
02 paid = input("Enter amount paid")
03 global change = paid - total
04 calculateChange()
05
06 procedure calculateChange()
07      twenty = 0
08      ten = 0
09      five = 0
10      while change >= 20 //Calculates number of £20 notes needed
11          twenty = twenty + 1
12          change = change - 20
13      endwhile
14      while change >= 10 //Calculates number of £10 notes needed
15          ten = ten + 1
16          change = change - 10
17      endwhile
18      while change >= 5 //Calculates number of £5 notes needed
19          five = five + 1
20          change = change - 5
21      endwhile
22      print("The amount of change you need is £" + str(change))
23      print("Total £20 Notes:" + str(twenty))
24      print("Total £10 Notes:" + str(ten))
25      print("Total £5 Notes:" + str(five))
26 endprocedure
```

(a) Describe how `calculateChange()` on line 04 is used differently to `calculateChange()` on line 06.

.....

.....

.....

..... [2]

(b) When line 22 is run, it will always print:

```
The amount of change you need is £0
```

Explain why this error occurs when line 22 is run.

.....

.....

.....

..... [2]

(c) Explain why Kylie has used `str` on lines 22 to 25 in her algorithm.

.....

.....

.....

.....

..... [3]

4* Anna currently writes her program code in a text editor and then runs the compiler.

She has been told that using an Integrated Development Environment (IDE) would be more helpful.

Discuss the benefits of Anna using an IDE to write and test her program rather than using a text editor. [9]

.....

.....

.....

.....

.....

.....

.....

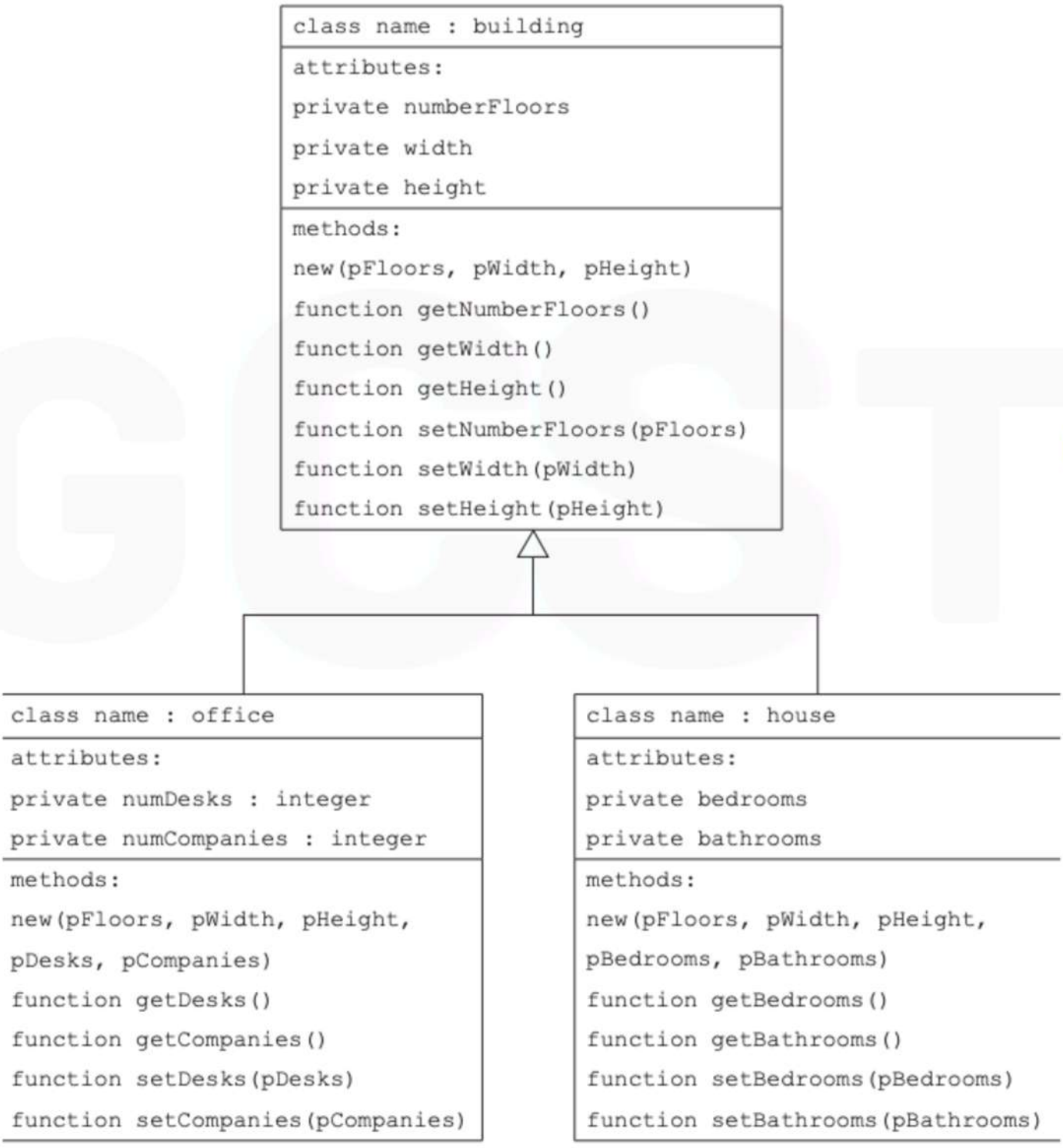
.....

.....

.....

.....

5 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:



(a) The method `new` is used to denote the constructor for each class.

State the purpose of the constructor.

.....

..... **[1]**

The classes `office` and `house` inherit from `building`.

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

.....

.....

.....

..... **[2]**

(c) 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]

(d) Write program code or pseudocode to declare the class `house`.

Define the attributes and constructor method in your answer. You do **not** need to write the get or set methods. [6]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(e) 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]

(f) Christoff wants to create a new `house` called `houseOne`. It has the properties: 2 floors, 8(m) width, 10(m) height, 3 bedrooms and 2 bathrooms.

The house is located on a road with the identifier `limeAvenue` of type `houseRoad`, `houseOne` is the first house in this road.

Write pseudocode or program code to declare the house `houseOne`, road `limeAvenue` and assign `houseOne` to the first array position in the road.

.....

.....

.....

.....

.....

.....

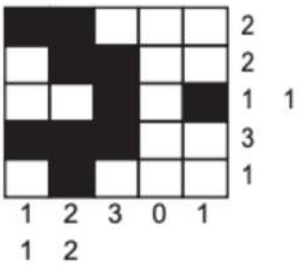
.....

.....

..... **[4]**

8 A Nonogram is a logic puzzle where a player needs to colour in boxes. The puzzle is laid out as a grid and each square needs to be either coloured black or left white.

The numbers at the side of each row and column tells the player how many of the boxes are coloured in consecutively. Where a row has two or more numbers, there must be a white square between the coloured squares.



In this example:

- the first column has 1 1, this means there must be two single coloured boxes in this column. There must be at least 1 white box between them.
- the first row has 2, this means there must be two consecutively coloured boxes in the row.

Juan is creating a program that will store a series of Nonograms for a user to play. The game will randomly select a puzzle and display the blank grid with the numbers for each row and column to the user.

The user plays the game by selecting a box to change its colour. If the box is white it will change to black and if it is black it will change to white. The user can choose to check the answer at any point, and the game will compare the grid to the answers and tell the user if they have got it correct or not.

(b) Juan uses the structure diagram to create a modular program with a number of subroutines. The program will use two integer 2-dimensional arrays to store the puzzles:

- `puzzle(5,5)` stores the solution
- `answerGrid(5,5)` stores the user's current grid.

A `0` represents a white box and a `1` represents a black box.

(i) Juan creates a function, `countRow()`, to count the number of coloured boxes in one row and return the number of consecutive coloured boxes in that row. If there is more than one set of coloured boxes in the row, these are joined together and the string is returned.

For example, in the following grid `countRow` for row 0 will return `"2"` as a string, and `countRow` for row 2 will return `"1 1"` as a string. If there are no 1s in a row, then `"0"` is returned as a string.

1	1	0	0	0
0	1	1	0	0
0	0	1	0	1
1	1	1	0	0
0	1	0	0	0

Complete the pseudocode algorithm `countRow()`.

```
01 function countRow(puzzle:byref, rowNum:byval)
02   count = 0
03   output = " "
04   for i = 0 to ……
05     if puzzle[rowNum, i] == …… then
06       count = count + 1
07     elseif count >= 1 then
08       output = output + str(……) + " "
09       count = 0
10     endif
11   next i
12   if count>= 1 then
13       output=output+str(count)
14   elseif output == "" then
15       output = "……"
16   endif
17   return ……
18 endfunction
```

[5]

(ii) Explain the purpose of line 03 in the function `countRow`.

.....

.....

.....

..... [2]

(iii) Describe the purpose of branching and iteration in the function `countRow`.

.....

.....

.....

.....

.....

..... [3]

(iv) The procedure `displayRowAnswer()` takes `puzzle` as a parameter and outputs the value in each box. Each box in a row is separated by a space. At the end of each row there are two spaces and (by calling the function `countRow` from **part 8(b)(i)**) the clue values for that row.

For example the puzzle below:

1	1	0	0	0
0	1	1	0	0
0	0	1	0	1
1	1	1	0	0
0	1	0	0	0

Would output:

```
1  1  0  0  0     2
0  1  1  0  0     2
0  0  1  0  1     1  1
1  1  1  0  0     3
0  1  0  0  0     1
```

Write pseudocode or program code for the procedure `displayRowAnswer()`. [6]

.....

.....

.....

.....

.....

.....

.....

(v) The function `checkWon()` takes `answerGrid` and `puzzle` as parameters and compares each element in the grids. If they are identical, it returns `true`, otherwise returns `false`.

```
01 function checkWon(puzzle)
02   for row = 0 to 4
03     for column = 0 to 4
04       if puzzle[row, column] == answerGrid[row, column] then
05         return false
06       endif
07     next column
08   next column
09   return true
10 endfunction
```

There are **three** logic errors in the function `checkWon`.

State the line number of each error and give the corrected line.

Error 1 line number .....

Error 1 correction .....

Error 2 line number .....

Error 2 correction .....

Error 3 line number .....

Error 3 correction .....

[3]

(c)* Juan passed the two arrays as parameters, but he did consider making them globally accessible.

Compare the use of global and local variables and data structures in this program. Include the use of parameters and program efficiency in your answer.

.... [9]

3 Hugh has written a recursive function called `thisFunction()` using pseudocode.

```
01  function thisFunction(theArray, num1, num2, num3)
02    result = num1 + ((num2 - num1) DIV 2)
03    if num2 < num1 then
04      return -1
05    else
06      if theArray[result] < num3 then
07        return thisFunction(theArray, result + 1, num2, num3)
08      elseif theArray[result] > num3 then
09        return thisFunction(theArray, num1, result - 1, num3)
10      else
11        return result
12      endif
13    endif
14  endfunction
```

The function `DIV` calculates integer division, e.g. 5 DIV 3 = 1

(a) `theArray` has the following data:

Index:	0	1	2	3	4	5	6	7
Data:	5	10	15	20	25	30	35	40

Trace the algorithm, and give the final return value, when it is called with the following statement:

```
thisFunction(theArray, 0, 7, 35)
```

You may choose to use the table below to give your answer.

.....

.....

.....

Function call	num1	num2	num3	result
thisFunction(theArray,0,7,35)				

Final return value ..... [5]

(b) State the name of the standard algorithm `thisFunction()` performs.

.....

..... [1]

(c) Hugh could have written `thisFunction()` using iteration instead of recursion.

Compare **two** differences between recursion and iteration.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

Rewrite the function `thisFunction()` so that it uses iteration instead of recursion.

You should write your answer using pseudocode or program code.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6]

4 The following pseudocode procedure performs an insertion sort on the array parameter.

```
01 procedure insertionSort(dataArray:byRef)
02   for i = 1 to dataArray.Length - 1
03     temp = dataArray[i]
04     tempPos = i - 1
05     exit = false
06     while tempPos >= 0 and exit == false
07       if dataArray[tempPos] < temp then
08         dataArray[tempPos + 1] = dataArray[tempPos]
09         tempPos = tempPos - 1
10       else
11         exit = true
12       endif
13     endwhile
14     dataArray[tempPos + 1] = temp
15   next i
16 endprocedure
```

(a) Explain why `dataArray` is passed by reference and not by value.

.....

.....

.....

..... **[2]**

(b) State whether the procedure `insertionSort` sorts the data into ascending or descending order and explain your choice.

.....

.....

.....

.....

.....

..... **[3]**

6 Barney is writing a program to store data in a linked list. He is writing the initial program for a maximum of 10 data items.

(b) Barney wants the nodes to be stored as objects using object-oriented programming. He designs the following class.

class: node
attributes: private data : Real private pointer : Integer
methods: new (newData, newPointer) getData() getPointer() setData(newData) setPointer(newPointer)

The constructor assigns the parameters to the attributes to create an object.

(i) Write an algorithm, using pseudocode or program code, to create the class `node`, its attributes and constructor.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(ii) The class `node`, uses get methods and set methods.

Describe **one** difference between get methods and set methods.

.....

.....

.....

..... **[2]**

(ii) Barney will use an Integrated Development Environment (IDE) to debug his program code.

Describe **three** features commonly found in IDEs that Barney could use to debug his program code.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

3 .....

.....

.....

.....

[6]

(e)* Barney would like his linked list to be part of a base program that is saved in a library. This means that it can be reused and changed by other programs.

Discuss the benefits of using different object-oriented techniques that Barney could use to achieve this. [12]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

1 Sally is a classroom teacher. She would like a program to be able to organise where students will sit in her classroom.

(b) Sally would like to increase the security of her program by adding a password to enter the program. She has created the procedure, `checkPassword`, to do this.

```
01 procedure checkPassword()
02     correctPassword = "ComputerScience12"
03     check = false
04     while check == false
05         enteredPassword = input("Enter Password")
06         if enteredPassword == correctPassword then
07             check = true
08         endif
09     endwhile
10 endprocedure
```

(i) Identify the programming construct used on lines 06 to 08 in the procedure `checkPassword`.

.....

..... **[1]**

(ii) Sally has used a `while` loop on line 04 of the procedure `checkPassword`.

Explain why Sally has used a `while` loop instead of a `for` loop.

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(iii) Sally could have used a `do until` loop instead of a `while` loop.

Rewrite lines 04 to 09 of the procedure `checkPassword` using a `do until` loop instead of a `while` loop.

.....

.....

.....

.....

.....

..... **[3]**

(c) Sally will make use of an Integrated Development Environment (IDE) to create her program code.

(i) Describe **three** features that are commonly found in IDEs that will help Sally write her program code.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

3 .....

.....

.....

.....

[6]

4 Daisy is a computer technician. She is responsible for making sure all new employees are given a username to access the computer network.

The rules that are followed when creating a new username are as follows:

Step 1: The employee's first name is entered (e.g. Roger)

Step 2: The employee's surname is entered (e.g. Banks)

Step 3: A username is then made up from:

- Their whole surname (e.g. Banks)
- The first letter of their first name (e.g. R)
- A number 1

For example: BanksR1

Step 4: The username is then checked against existing usernames. This is done by calling a function `existingUsers`. This will return `true` if the username is unique and `false` if the username already exists.

Step 5: If the username is unique then "Username is Unique" should be printed. If the username already exists then the number at the end of the username should increase by one (e.g. BanksR2).

Step 6: Steps 4 and 5 should be repeated until the username is unique.

Write a procedure called `createUsername` that meets the rules of Daisy's program.

You should write your procedure using pseudocode or program code.

..... [9]

GCST

2019

3 A recursive function, GCD, is given in pseudocode.

```
function GCD(num1, num2)
        if num2 == 0 then
                return num1
        else
                return GCD(num2, num1 MOD num2)
        endif
endfunction
```

(a) The function uses branching.

(i) Identify the type of branching statement used in the function.

..... **[1]**

(ii) Explain the difference between branching and iteration.

.....

.....

.....

..... **[2]**

(iii) Identify the **two** parameters in the function.

1 .....

2 .....

[1]

(iv) State whether the parameters should be passed by value, or by reference. Justify your answer.

.....

.....

.....

..... **[2]**

(v) Describe the arithmetic operation of `MOD`. Use an example in your answer.

.....

.....

.....

.....

.....

..... [2]

(b) Trace the recursive function when it is called by the statement `GCD(250, 20)`. Give the final value returned.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Final return value:.....

[3]

(c) The function has been rewritten using iteration instead of recursion.

(i) State **one** benefit and **one** drawback of using iteration instead of recursion.

Benefit.....

.....

Drawback.....

.....

[2]

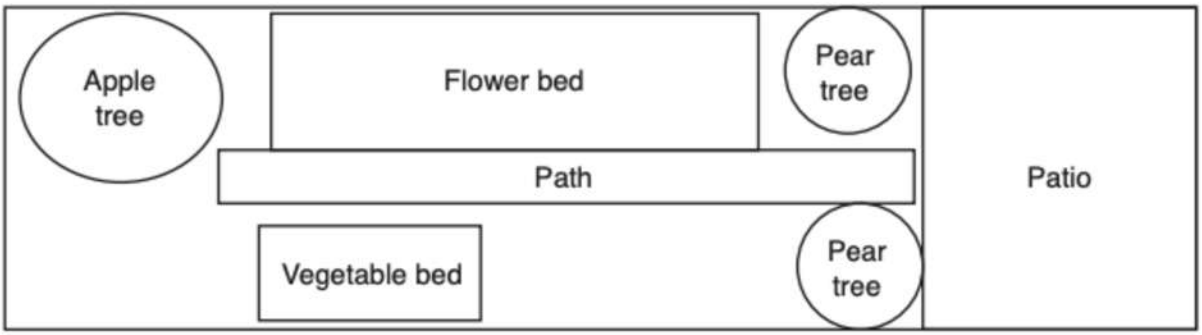
(ii) Complete the missing statements in this iterative version of the function.

```
function newGCD(num1, num2)
  temp = 0
  while (num2 != ..........)
    .......... = num2
    num2 = num1 MOD ..........
    num1 = temp
  endwhile
  return ..........
endfunction
```


[4]

7 A program is needed to plan the layout of a garden.

The program will allow the user to create an image of the garden, for example:



(b) The program is to be built using object oriented programming.

All items that can be added to the garden are declared as instances of the class `GardenItem`.

The class has the following attributes:

Attribute	Description	Example
`itemName`	The name of the item	Flowerbed
`length`	The length of the item in metres	2
`width`	The width of the item in metres	1

(i) The constructor method sets the attributes to values that are passed as parameters.

Write pseudocode or program code to declare the class `GardenItem` and its constructor. All attributes should be private and initialised through the constructor (e.g. `daisies = new GardenItem("Flowerbed",2,1)`). **[4]**

.....

.....

.....

.....

.....

.....

(ii) The trees in the garden layouts are defined by the class `Tree`. This class inherits from `GardenItem`.

The class `Tree` has the additional attributes: `height`, `sun`, `shade`.

If `sun` is `true` then the tree can grow in full sun, if it is `false` then it cannot.

If `shade` is `true` then the tree can grow in full shade, if it is `false` then it cannot.

The length and width of a tree are the same. Only one value for these measurements is passed to the constructor.

Write an algorithm, using pseudocode or program code, to declare the class `Tree`. Declare all attributes as private.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

(iii) The Common Oak is a type of tree. It has a maximum height, length and width of 40 m. It can grow in sun and shade.

Write a statement, using pseudocode or program code, to declare an instance of tree for the Common Oak. Give the object the identifier `firstTree`.

.....

.....

..... [4]

(iv) The classes `GardenItem` and `Tree` use get and set methods to access and alter their private attributes.

Write the get method `getItemName` and set method `setItemName` for class `GardenItem`. The set method takes the new value as a parameter.

Do not write any other methods, or re-declare the class.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(v) The trees in the garden layouts are stored in a 1-dimensional array, `treeArray`. The array can store a maximum of 1000 items. The array has global scope.

A procedure, `findTree`, takes as parameters:

- The maximum height of a tree
- The maximum width of a tree
- Whether the tree can live in full sun
- Whether the tree can live in full shade.

It searches the array, `treeArray`, for all trees that do not exceed the maximum height and width, and that can grow in the conditions available. If there are no suitable trees, a suitable message is output.

It outputs the name and details of the trees found in an appropriate message.

Call the get methods, `getItemName`, `getHeight`, `getWidth`, `getSun`, `getShade`, to access the attributes.

Write, using pseudocode or program code, the procedure `findTree`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

2 A procedure is shown in the following pseudocode.

The arrays that are passed to the procedure store integer values.

`length` returns the total number of elements the array can hold.

```
01 procedure calculateOnce(data[]:byRef, nextData[]:byRef)
02       if data.length > nextData.length then
03                loopCount = nextData.length - 1
04       else
05                loopCount = Data.length - 1
06       endif
07       count = 0
08       while count <= loopCount
09                data[count] = data[count] + nextData[count]
10                count = count + 1
11       endwhile
12 endprocedure
```

(a) A decision is made on line `02`.

(i) Identify the line where the second decision is made.

..... **[1]**

(ii) Explain the purpose of the code in lines `02` to `06`.

.....

.....

.....

.....

.....

..... **[3]**

(b) The procedure has parameters passed by reference.

(i) Give the identifiers of the **two** parameters.

1 .....

2 .....

[2]

(ii) State the effect of the array `data[]` being passed by reference and not by value.

.....

..... **[1]**

4 A program corrects the grammar in a line of text. The text is read in from a text file.

(a) The function, `getText`, needs to:

- take the file name as a parameter
- open the file
- read the line of data in the text file into one string
- return the string of data.

Write the function `getText`. [4]

.....

.....

.....

.....

.....

.....

(b) The procedure, `fullStop`, needs to:

- ask for a file name as input
- read the data from the file using the function `getText`
- replace the first letter after each full stop with a capital letter if it is currently lower case (if the next character is a space, it must check each successive character until it finds a letter)
- write the edited data back to the text file.

You can assume the text file only contains upper and lower case letters, spaces and full stops.

Part of the ASCII table has been provided:

ASCII Value	Character
65	"A"
90	"Z"
97	"a"
122	"z"
32	" " (space)
46	"." (full stop)

The following functions may be used in your answer:

`asc(character)` returns the ASCII value for a single character, e.g. `asc("A")` would return 65.

`upper(character)` returns the single character in upper case, e.g. `upper("a")` would return "A".

Write the procedure `fullStop`. **[7]**

.....

.....

.....

.....

.....

.....

.....

4 A recursive function, `generate`, is shown.

```
function generate(num1:byval)
    if num1 > 10 then
        return 10
    else
        return num1 + (generate(num1 + 1) DIV 2)
    endif
endfunction
```

(a) Trace the algorithm to show the value returned when `generate(7)` is called. Show each step of your working.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

(b) The parameter, `num1`, is passed by value.

Explain why the parameter was passed by value instead of by reference.

.....

.....

.....

..... **[2]**

(c)* Parameters can be used to reduce the use of global variables.

Compare the use of parameters to global variables in recursive functions.

.... **[9]**

(d) A student called Jason writes a recursive algorithm. The recursive algorithm uses more memory than if Jason had written it as an iterative algorithm.

Explain why the recursive algorithm uses more memory than the iterative algorithm.

.... **[2]**

5 A computer program stores data input on a stack named `dataItems`. The stack has two sub-programs to add and remove data items from the stack. The stack is implemented as a 1D array, `dataArray`.

(iii) The array has 50 items.

The function, `searchItem()`, performs a linear search for a data item.

```
function searchItem(dataItem)
    for count = 0 to 49
        if dataArray[count] == dataItem then
            return(count)
        endif
    next count
    return(-1)
endfunction
```

Rewrite the function using a while loop.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

1 A user enters whole numbers into a computer program. Each number entered is placed onto a stack. The stack is created using an array with a maximum of 20 elements.

(ii) The parameter can be passed by value or by reference.

Describe what is meant by passing a parameter by value and by reference.

By value .....

.....

.....

.....

By reference .....

.....

.....

.....

[4]

(iv) The procedure, `calculate`, takes each item in turn from the stack. It alternately adds then subtracts the numbers until there are none left.

For example, if `numStack` contains:

2
6
5
12

It would perform 2 + 6 – 5 + 12 and output 15.

```
01    procedure calculate()
02          total = 0
03          add = true
04          if top == 0 then
05                print("Stack empty")
06          else
07                total = numStack[top - 1]
08                top = top - 1
09                while top != 0
10                      if add == true then
11                            total = total + numStack[top - 1]
12                            add = false
13                      else
14                            total = total - numStack[top - 1]
15                            add = true
16                      endif
17                      top = top - 1
18                endwhile
19                print(total)
20          endif
21    endprocedure
```

Complete the trace table for the procedure `calculate`. The current array and pointer values when the procedure is called are on the first line of the trace table.

	numStack								
top	0	1	2	3	4	5	total	add	Output
5	20	2	6	12	8				

[6]

2 A games company has developed a game called Kidz Arrowz. The players throw an arrow at a target board and are awarded different points depending on which circle the arrow lands. Fig. 1 shows the board.

(e) The programmer uses an Integrated Development Environment (IDE) to develop the program.

Describe how the IDE could be used to create the Kidz Arrowz program.

.....

.....

.....

.....

.....

..... **[3]**

2 A programmer is developing an ordering system for a fast food restaurant. When a member of staff inputs an order, it is added to a linked list for completion by the chefs.

(c) The linked list is implemented using a 2D array, `theOrders`:

- Row 0 stores `orderNo`
- Row 1 stores `next`

The data now stored in `theOrders` is shown in Fig. 2.2.

184	186	185	187
1	2	3	

Fig. 2.2

`theOrders[1,0]` would return `1`

The following algorithm is written:

```
procedure x()
    finished = false
    count = 0
    while NOT(finished)
        if theOrders[1,count] == null then
            finished = true
        else
            output = theOrders[0,count]
            print(output)
            count = theOrders[1,count]
        endif
    endwhile
    output = theOrders[0,count]
    print(output)

endprocedure
```

(ii) Complete the trace table for procedure `x`, for the data shown in Fig. 2.2.

finished	count	output

[3]

(e) The programmer is writing the program using an IDE.

Identify **three** features of an IDE that the programmer would use when writing the code and describe how the features benefit the programmer.

1.....

.....

.....

.....

2.....

.....

.....

.....

3.....

.....

.....

.....

[6]

5 A recursive function, `calculate`, is shown below:

```
01 function calculate(num1, num2)
02     if num1 == num2 then
03         return num1
04     elseif num1 < num2 then
05         return calculate(num1, (num2-num1))
06     else
07         return calculate(num2, (num1-num2))
08     endif
09 endfunction
```

(a) Identify the lines where recursion is used.

..... **[1]**

(b) Trace the algorithm, showing the steps and result when the following line is run:

```
print(calculate(4,10))
```

[5]

(c) Re-write the function so it uses iteration instead of recursion.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

6 A software developer is creating a Virtual Pet game.

The user can choose the type of animal they would like as their pet, give it a name and then they are responsible for caring for that animal. The user will need to feed, play with, and educate their pet.

The aim is to keep the animal alive and happy, for example if the animal is not fed over a set period of time then the pet will die.

- The game tells the user how hungry or bored the animal is as a percentage (%) and the animal's intelligence is ranked as a number between 0 and 150 (inclusive).
- Hunger and boredom increase by 1% with every tick of a timer.
- When the feed option is selected, hunger is reduced to 0.
- When the play option is selected, bored is reduced to 0.
- When the read option is selected, the intelligence is increased by 0.6% of its current value.

An example of the game is shown:

```
What type of pet would you like? Fox or Elephant?
Fox
What would you like to name your Fox?
Joanne
Joanne's stats are
Hunger: 56%
Bored: 85%
Intelligence: 20
What would you like to do with your pet? Play, Read or Feed?
```

(c) The developer needs to write procedures for the options play and read. Each of the options changes its corresponding value, and outputs the results to the screen.

(i) Write a procedure, using pseudocode, to reset `bored` and output the new value in an appropriate message.

.....

.....

.....

.....

.....

..... **[3]**

(ii) Write a procedure, using pseudocode, to increase `intelligence` by 0.6% and output the new intelligence in an appropriate message.

.....

.....

.....

.....

.....

..... **[3]**

(d) The developer is extending the game to allow users to have multiple pets of different types. The developer has written a class, `Pet`.

The attributes and methods in the class are described in the table:

Identifier	Attribute/Method	Description
`petName`	Attribute	Stores the pet's name
`bored`	Attribute	Stores the % bored
`hunger`	Attribute	Stores the % hunger
`intelligence`	Attribute	Stores the intelligence
`type`	Attribute	Stores the type of animal
`new`	Method	Creates a new instance of `pet`
`feed`	Method	Reduces `hunger` to 0 and outputs `hunger`
`play`	Method	Reduces `bored` to 0 and outputs `bored`
`read`	Method	Increases `intelligence` by a set value
`outputGreeting`	Method	Outputs a message to the user

Part of the class declaration is given:

```
class Pet
    private petName
    private bored
    private hunger
    private intelligence
    private type
...
...
```

(i) After a user enters the pet name, and chooses a type, the constructor method of `Pet` is called to create a new instance. The method needs to set `petName`, as well as `hunger`, `bored` and `intelligence` to starting values of 0.

Write, using pseudocode, the constructor method for this class.

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(ii) Write a line of code that creates a new instance of `Pet` for a Tiger called "Springy".

..... **[2]**

(iii) The method `outputGreeting` for the superclass is written as follows:

```
public procedure outputGreeting()
    print("Hello, I'm " + petName + ", I'm a " + type)
endprocedure
```

A class is needed for Tiger. The class needs to:

- inherit the methods and attributes from `pet`
- in the constructor, set `type` to Tiger, `intelligence` to 10, `hunger` to 50 and `bored` to 10
- extend the method `outputGreeting`, by outputting an additional line that says "I like to eat meat and roar"

Write, using pseudocode, the class `Tiger`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

GCST

2016

AS - Level

1 Programming languages consist of three basic programming constructs. For each construct, state its name and give a working example.

Construct 1: .....

Example: .....

.....

.....

.....

Construct 2: .....

Example: .....

.....

.....

.....

Construct 3: .....

Example: .....

.....

.....

.....

[6]

2* A software company is going to write a policy on its use of variables when writing programs. The policy will cover the use of local and global variables and variable naming rules and conventions.

Discuss the content of this policy and why it is required. **[9]**

6 **(a)** A programmer is going to design a procedure that will prompt for and receive two values, A and B. The procedure will then compare them. The procedure will also write a suitable message to a file on disk depending on whether:

- the values are the same
- A is less than B, or
- B is less than A.

Use pseudocode to write the procedure. [5]

4.

(i) A procedural programming language may use procedures.

Explain the term procedural programming language.

---[2]

(ii) The same variable name may be used in more than one procedure in a program.

Explain how a variable named result may be used in different procedures without causing errors.

---[2]

(iii) Explain parameter passing.

---[5]

5. Variables are used in programming.

(i) Describe the use of local variables.

---

[4]

(ii) State **two** features of global variables that distinguish them from local variables.

1

2

---

[2]

6. A variable can be declared as global or local and is said to have scope.

(i) Explain what is meant by the term 'variable'.

---

[2]

(ii) Explain what is meant by 'scope' in relation to global and local variables.

---

[2]

7. A company employs a systems analyst to create a program for calculating wages. The data that he analyses has the following items:

- Hourly rate.
- Hours worked.
- Tax paid.
- National Insurance paid.

Various variables are used in the execution of the program.

Explain the term 'scope' in relation to variables within a program that calls several different procedures.

---

[6]

14. A user enters whole numbers into a computer program. Each number entered is placed onto a stack. The stack is created using an array with a maximum of 20 elements.

Part of the array, `numStack`, is shown when one number has been input.

top	1

index	stackItem
9	
8	
7	
6	
5	
4	
3	
2	
1	
0	20

The pointer, `top`, points to the next free space in the stack.

A function, `addItem`, takes a number as a parameter and adds the number to the stack. The function returns `true` if this was successful, and `false` if the stack is already full.

(i) Give **one** reason why a function is used instead of a procedure in this scenario.

---

[1]

(ii) The parameter can be passed by value or by reference.

Describe what is meant by passing a parameter by value and by reference.

By value ---

By reference ---

---

[4]

```
01  procedure calculate()
02      total = 0
03      add = true
04      if top == 0 then
05          print("Stack empty")
06      else
07          total = numStack[top - 1]
08          top = top ? 1
09          while top != 0
10              if add == true then
11                  total = total + numStack[top - 1]
12                  add = false
13              else
14                  total = total ? numStack[top - 1]
15                  add = true
16              endif
17              top = top - 1
18          endwhile
19          print(total)
20      endif
21  endprocedure
```

Complete the trace table for the procedure `calculate`. The current array and pointer values when the procedure is called are on the first line of the trace table.

top	numstak						total	add	output
	0	1	2	3	4	5			
5	20	2	6	12	8				

[6]

20. Dexter is leading a programming team who are creating a computer program that will simulate an accident and emergency room to train hospital staff.

Dexter's team is using an integrated development environment (IDE).

Describe how the programmers could make use of the following IDE tools:

Breakpoints _____

_____

_____

_____

Stepping _____

_____

_____

_____

[4]

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

THANK YOU!

GCST

GCSECOMPUTERSCIENCETUTOR.COM