Year 11 Computer Science Programming Theory Test - Sample Questions

Materials required/recommended for this paper

To be provided by the supervisor

This Question/Answer Booklet

To be provided by the candidate

Standard items: pens (blue/black preferred), pencils (including coloured), sharpener, correction tape/fluid, eraser, ruler, highlighters

Special items: non-programmable calculators approved for use in this examination, Mathomat and/or Mathaid and/or any system flowchart template

Question 1 **(9 marks)**

Vikki would like a program to help her work out how many drinks she can buy from the school canteen. Drinks cost $2 each.

```
MODULE Main
        INPUT(amount)
        IF amount >= 2 THEN
                OUTPUT("You can afford 1 drink")
        ELSE IF amount >= 4 THEN
                OUTPUT("You can afford at least 2 drinks")
        ELSE
                OUTPT("You can't afford any drinks")
END Main
```

(a) Identify two errors in the above algorithm. For each error, identify the type of error and explain why it is an error. (6 marks)

1 mark each error identified correctly
1 mark each error type identified correctly
1 mark each explanation of error

Errors include:
Error 1: On line 8, OUTPT is incorrect
Type: Syntax error
Explanation: The command OUTPT is not a valid commend, it should be OUTPUT

Error 2: The IF and ELSE…IF conditions are the wrong way around
Type: Logic error
Explanation: The way the pseudocode is entered, any value greater than 2 will tell the user they can only receive 1 drink as the first IF condition is amount >= 2. The amount >= 4 should be first, then amount >= 2, then ELSE

Error 3: The IF statement does not have an END IF
Type: Syntax error
Explanation: The IF statement needs and END IF to let the reader know when the IF control structure finishes.

(b) To be able to test her algorithm, Vikki would like to develop a test plan. Identify values that Vikki should use to test her algorithm and explain why she should use those values. (3 marks)

1 mark for each valid value that has been justified

For example:
Vikki needs to select test data that will test all the conditions and ensure that each line of code is tested. To do this she should select:
1 – to test for condition where not enough money
2 – to test boundary condition for enough money for 1 drink
4 – to test boundary condition for enough money for more than 1 drink

Use the following information to answer Questions 2 and 3
Ellyse has recently installed some new machinery in her warehouse and needs to check the heat that is being produced to ensure a safe working environment. She has set up a system to read the temperature each hour and store that temperature in an array.

Before anyone is allowed to enter the warehouse to work, Ellyse is required to check that the first ten (10) readings are within acceptable limits. The temperature gauge reads data in Celsius and the initial readings are:

temperatures ← [40.5, 45, 34, 43, 48, 51.3, 37, 38, 46.7, 49]

Question 2 (3 marks)

(a) Explain why an array is the most suitable data structure to store the temperature data. (2 marks)

1 mark referencing ease of accessing multiple values related to same thing
1 mark discussing all data is of the same type

For example:
An array is the most suitable data type as it allows the programmer to store multiple values of the same data type in one memory structure. This makes it easier for the programmer to access the values and makes the code much neater.

(b) Identify the type of data that will be stored in the *temperatures* array. (1 mark)

1 mark correct type

Real/Floating point
NOTE: Only accept these as they are what are in the syllabus

Question 3 (7 marks)

Write the function *AverageTemperature* which will take in the temperatures array as a parameter and returns the average temperature after the initial 10 readings.

```
FUNCTION AverageTemperature(temperatures)
        total ← 0
        FOR i ← 0 TO 9 DO
                total ← total + temperatures[i]
        END FOR
        AverageTemperature ← total / 10
END AverageTemperature
```

NOTE: Accept using temperatures.length to calculate length of loop and average

1 mark FUNCTION/END
1 mark parameters
1 mark initialising total
1 mark looping through array correctly (note correct indexing starting at 0)
1 mark calculating total correctly
1 mark calculating average
1 mark returning average

Use the following information to answer Questions 4 and 5

Jake runs a business that allows customers to hire garden equipment such as lawn mowers and hedge trimmers. He wants to develop an application to simplify the process of calculating the cost of hiring different machines and process the customer payments. Jake has written the pseudocode below.

```
1       MODULE Main
2           hourlyCost ← 0
3           machine ← “”
4           hours ← 0
5           cost ← 0

6           OUTPUT(“What type of machine are you hiring?”)
7           INPUT(machine)
8           CALL FindMachineCost(machine, hourlyCost)

9           REPEAT
10              OUTPUT(“How many hours do you need the machine?”)
11              INPUT(hours)
12          UNTIL hours > 0 and hours <= 10

13          cost ← CostOfHire(hours, hourlyCost)
14          OUTPUT(“The machine will cost “ + cost + “ to hire.”)
15          OUTPUT(“Would you like to proceed? (Y/N)”)
16          INPUT(proceed)

17          IF proceed = ‘Y’ THEN
18              CALL ProcessPayment(cost)
19          ELSE
20              OUTPUT(“Thank you for your enquiry.”)
21          END IF
22      END Main

23      FUNCTION CostOfHire(hours, hourlyCost)
24          CostOfHire ← hours * hourlyCost
25      END CostOfHire

26      MODULE FindMachineCost(machine, hourlyCost)
27          hourlyCost ← 0
28          CASE machine OF
29              ‘lawnmower’    :      hourlyCost ← 5
30              ‘hedge trimmer’ :     hourlyCost ← 4
31              ‘whipper snipper’ :   hourlyCost ← 6
32              ‘leaf blower’ :       hourlyCost ← 2
33          END CASE
34      END FindMachineCost

35      MODULE ProcessPayment(cost)
36          OUTPUT(‘Your payment has been processed successfully’)
37      END ProcessPayment
```

Question 4 **(3 marks)**

Discuss how the use of modularisation will benefit Jake when designing his application.

3 marks full discussion of how modularisation could benefit Jake, including at least two factors of benefit
2 marks discussion of benefits of modularisation
1 mark statement of a benefit of modularisation

For example:
Modularisation significantly improves the process of developing software. It allows Jake to break the initial problem down into several smaller components that individually are more manageable. Each smaller component can be worked on independently of the others and solve a specific, small part of the overall task. For a large project, it also makes it easier for multiple people to work on the same project as they can work on individual module. Finally, modularisation makes the code easier to read and understand as well as making the code re-usable both within the project and within other projects.

Question 5 **(5 marks)**

(a) Identify the type of control structure that is being used on line 9 of the pseudocode above.

(1 mark)

Test-last loop (also accept REPEAT UNTIL loop).

NOTE: Must mention the **type** of iteration

(b) Explain the purpose of this section of the program. (2 marks)

2 marks correct explanation for purpose of code
1 mark partial explanation

For example:
This section of the program is designed to ask the user to enter the number of hours and perform a range check on the input to make sure that it is an acceptable value. The code will keep asking the user to enter a value until a valid value is entered, thus ensuring the validity of the input.

(c) Discuss why the control structure you have identified is the most appropriate control structure for this section of the program. (2 marks)

2 marks complete discussion including reference to this project
1 mark statement about appropriateness of control structure

For example:
In this example, the user needs to enter the number of hours, so this piece of code needs to run at least once. Using a loop ensures that the program will not progress until a valid number of hours has been entered. Since a test-last loop must run at least once, and will continue running until the required condition is met, then this is the most appropriate control structure to use.

Question 6 **(9 marks)**

Vikki wants to raise some money for the local primary school. To do this, she is running a silent auction. In this type of auction, each person enters a bid, and the person with the highest bid at the end gets to take home to the prize.

As she is busy, she wants a program that will help her determine the winner of the auction.

Using pseudocode, write a module *SilentAuction* that will:

- Ask the user to enter the name of the person making a bid until no name is entered
- Once a name is entered, ask for the amount of their bid
- Record the name of the person with the highest bid. If two people have the same bid, the first person with that bid is the winner
- Print the winner of the auction

For example, the program should work something like the following:

Name: Peter
Bid: 150
Name: Max
Bid: 100
Name: Jane
Bid: 155
Name:
The winner is Jane

```
MODULE SilentAuction
        winner ← ' '
        highbid ← -1

        bidder ← ' '
        bid ← ' '

        INPUT(bidder)
        WHILE bidder != ' '
                INPUT(bid)
                IF bid > highbid THEN
                        winner ← bidder
                        highbid ← bid
                END IF
                INPUT(bidder)
        END WHILE

        OUTPUT(bidder)
END SilentAuction
```

1 mark MODULE definition (including END Module)
1 mark initialise variables (winner and highbid)
1 mark input bidder
1 mark continue entering bidder until no bidder entered
1 mark input bid
1 mark checking if new bid is higher than existing bid
1 mark keeping track of current winner
1 mark keeping track of current highest bid
1 mark outputting winner

Question 7 **(5 marks)**

Margaret is employed as a software developer at Google where she works on the team that maintains Gmail. In her spare time, often during her lunch break at work, she works on her own private projects, such as developing an online photo storage tool. She has written most of the code herself and is planning on selling her new tool in time for Christmas. There are several sections of the code that she had problems with, however, and she used her position as a developer to have a look at the code base of Google Photos to get some ideas (although she was careful not to directly copy the code).
Discuss the legal and ethical issues associated with Margaret's photos storage tool.

5 marks detailed discussion of 3 issues (at least 1 legal and at least 1 ethical)
4 marks detailed discussion of 2 issues (1 legal and 1 ethical) or comment about 4 issues (2 legal and 2 ethical)
3 marks detailed discussion of 1 issue and comment on 1 issue or comment about 3 issues (at least 1 legal and 1 ethical)
2 marks detailed discussion of 1 issue or comment about 2 issues (1 legal and 1 ethical)
1 mark comment about 1 issue

Legal issues include:

- Google owns copyright to work done on their computers
- Google owns copyright to all the code base of Google Photos

Ethical issues:

- Margaret should not be working on her own code that she plans to sell while she is at work
- Margaret should not be looking at the Google code base without the permission of the authors
- Margaret is violating the trust given to her by Google as a developer by using her position to help produce her own product that will be in competition with Google

Question 8 **(4 marks)**

Jake is learning to play darts but has trouble working out his score after each turn.

The maximum score that can be achieved with three darts (one turn) is 180, and he wants to know:

- His score for this turn
- How far short of the maximum score he is

Using pseudocode, write a module that will:

- Ask the user to enter the three numbers – the score for each dart (this will be a number between 0 and 60)
- Calculate Jake's total score for this turn
- Print out the total score
- Calculate how far short of the maximum score he is
- Print out how far short of the maximum score he is

For example, the program should work something like the following:

Dart 1: 30
Dart 2: 18
Dart 3: 4
You scored 52 this round.
That is 128 below the maximum.

```
MODULE Main
        INPUT(score1)
        INPUT(score2)
        INPUT(score3)

        total ← score1 + score2 + score3
        difference ← 180 – total

        OUTPUT(total)
        OUTPUT(difference)
END Main
```

1 mark input three scores
1 mark calculate total
1 mark calculate difference
1 mark output total and difference

Question 9 **(4 marks)**

James has found some software online that is designed to allow him to remotely connect to another device to share files between devices. This software is aimed at people working from home who would like to share files with their colleagues. The developer provides a link to download the software, but clearly states on his website that the user should pay a fee before downloading and using the software.

(a) Describe one legal issue that James should consider when using this software. (2 marks)

2 marks for appropriate legal issued described in context of scenario
1 mark for legal issue identified

For example:
James needs to ensure that he follows appropriate copyright issues and pays for the software prior to using it. As the developer has explicitly stated that users must pay for the software, if James downloads the software and uses it without payment then he is infringing copyright.

Other legal issues could include:

- Do not connect to other person's device without permission

(b) Describe one ethical issue that James should consider when using this software. (2 marks)

2 marks for appropriate legal issued described in context of scenario
1 mark for legal issue identified

For example:
James should ensure that he uses the software for the purpose that it has been designed for, to share work files with colleagues. James should not use the software for other purposes, such as sharing music and movies between devices or using the software to remotely connect to another device and steal files or upload malicious software.

Question 10 **(5 marks)**

Consider the pseudocode below.

```
MODULE Main
    numbers ← [5, 7, 3, 4]
    FOR i ← 0 TO 3 DO
        IF i MOD 2 = 0 THEN
            numbers[i] ← numbers[i] + numbers[i + 1]
        ELSE
            numbers[i] ← numbers[i] + numbers[i – 1]
        END IF
    END FOR
END Main
```

NOTE: the operator MOD returns the remainder when the first number is divided by the second number. For example, 6 MOD 3 will return 0, while 5 MOD 4 will return 1

i	i MOD 2	i MOD 2 = 0	Numbers			
			0	**1**	**2**	**3**
			5	7	3	4
0	0	T	12			
1	1	F		19		
2	0	T			7	
3	1	F				11

2 marks array elements updated correctly
1 mark i column
1 mark i MOD 2 column
1 mark i MOD 2 = 0 column

Question 11 **(18 marks)**

Consider the following program used to calculate housing loans.

```
MODULE Main
        StampDuty ← 0
        INPUT(HousePrice)
        CALL CalcStampDuty(HousePrice, StampDuty)
        TotalHousePrice ← HousePrice + StampDuty + CalcSettlementFee(HousePrice)
        CALL CalcLoanAmount(TotalHousePrice)
        OUTPUT(LoanAmount)
END MODULE

MODULE CalcStampDuty(HousePrice, StampDuty)
        StampDuty ← 0.1*HousePrice
END MODULE

FUNCTION CalcSettlementFee(HousePrice)
        RETURN HousePrice * 0.05
END FUNCTION

MODULE CalcLoanAmount(TotalHousePrice)
        INPUT(Deposit)
        LoanAmount ← TotalHousePrice – Deposit
END MODULE
```

(a) Identify the **two (2)** local variables in this program outside of the Main module. (2 marks)

1 mark LoanAmount
1 mark Deposit

(b) State **two (2)** reasons why it is often not considered a good idea to use global variables.

(2 marks)
1 mark for each valid reason. Reasons may include:

- Modules that reference global variables cannot be reused in another program unless the global variable is also taken with them. This reduces their portability.
- If more than one module modifies a global variable it can become difficult to determine which module is causing a problem when errors relating to the global variable occur.
- When programming in a team, other programmers may modify a global variable without realising that some modules are relying on its value, which may result in errors

(c) An error occurs when the program tries to output LoanAmount. Explain the reason this is occurring and suggest one way the error could be corrected. (4 marks)

Explanation:
3 marks - explains that the variable LoanAmount exists within the local scope of the CalcLoanAmount module and therefore cannot be referenced outside of the module itself.
2 marks – explains that this is because LoanAmount is a local variable
1 mark – makes a superficial comment about LoanAmount

1 mark for a valid suggestion. May include:

- Move the LoanAmount output to the CalcLoanAmount module
- Return the value for LoanAmount from CalcLoanAmount and output the return value
- Make LoanAmount a global variable or pass it in as a reference parameter.

Question 12 **(7 marks)**

Consider the following program.

```
BEGIN
        INPUT(Num1)
        INPUT(Num2)
        WHILE Num1 != Num2 DO
                IF Num1 > Num2 THEN
                        Num1 ← Num1 - Num2
                ELSE
                        Num2 ← Num2 - Num1
                END IF
        END WHILE
        OUTPUT(Num1,Num2)
END
```

(a) Complete the trace table for the program below using the following pair of inputs as test data.

(5 marks)

Num1 ← 15, Num2 ← 41

Num1	Num2	Num1 != Num2	Num1 > Num2	Output
15	41	True	False	
	26	True	False	
4	11	True	True	
	7	True	False	
	3	True	False	
1		True	True	
	2	True	False	
	1	True	False	
		False		1,1

1 mark for each correct column including headings.

Note: Accept minor offsets/deviations from compressed form order if values are correct.

(b) Identify and describe the type of error that trace tables are used to detect. (2 marks)

1 mark identifies logic error
1 mark accurate description of a logic error

Example:
Logic errors occur when a program produces unexpected output or behaviour for a given input.

Question 13 **(3 marks)**

(a) Explain the term 'modularisation' as it applies to programming. (1 mark)
1 mark correct explanation

Example:
Modularisation is the process of breaking a problem into smaller subtasks.

(b) Describe two advantages of using modularisation in a program. (2 marks)

1 mark for each valid reason (max 2 marks)

Reasons could include:

- Improved readability of code
- Easier maintenance of code
- Easier debugging
- Ability to reuse code
- Ability for code to be developed by teams, with each team member working on a separate module

Use the following information to answer question 14.

Airlines are required to display a video to passengers informing them of the various risks and protocols associated with travelling during the pandemic. Big Red Roo Airlines have responded by producing a series of videos tailored to different age groups.

The module DisplaySafetyVideo selects and displays a video based on the string value returned from the BiggestAgeGroup function.

```
MODULE DispaySafteyVideo()
        Video ← default_video
        SelectedAgeGroup ← BiggestAgeGroup()
        CASE SelectedAgeGroup OF
                "elderly":      Video ← high_risk_video
                "adult":        Video ← med_risk_video
                "child":        Video ← low_risk_video
        END CASE
        OUTPUT(Video)
END MODULE
```

Question 14 **(7 marks)**

Complete the pseudocode for the BiggestAgeGroup function in the space provided below such that it returns a string, either "child", "adult" or "elderly", representing the age group with the largest number of passengers on the flight. Assume that the age groups will never be equal in number.

```
FUNCTION BiggestAgeGroup()
        AgeGroupTotals ← [0, 0, 0]
        FOR i ← 0 TO passengers.Length – 1
                CASE passengers[i].Age OF
                        >= 65: AgeGroupTotals[2] ← AgeGroupTotals[2] + 1
                        >= 18: AgeGroupTotals[1] ← AgeGroupTotals[1] + 1
                        < 18: AgeGroupTotals[0] ← AgeGroupTotals[0] + 1
                END CASE
        END FOR
        max_index ← 0
        max ← AgeGroupTotals[0]
        FOR i ← 1 TO 2
                IF AgeGroupTotals[i] > max THEN
                        max ← AgeGroupTotals [i]
                        max_index ← i
                END IF
        END FOR

        BiggestAgeGroup ← “child”

        CASE max_index OF
                = 0: BiggestAgeGroup ← “child”
                = 1: BiggestAgeGroup ← “adult”
                = 2: BiggestAgeGroup ← “elderly”
        END CASE

END FUNCTION
```

1 mark initialises variables used in calculation
1 mark determines suitable initial max total, such as the total at index 0 or a negative number.
1 mark loops through remaining values in the AgeGroupTotals array
1 mark checks if the current total in the array is greater than the previously held value each loop
1 mark accesses values in array appropriately by index using loop variable
1 mark correctly determines the index of the maximum total
1 mark selects the appropriate return value based on the index of the maximum AgeGroupTotal

Accept alternative solutions that arrive at a correct value such as algorithms using selection with either nesting or multiple conditions.

Sample alternative solution

```
BiggestAgeGroup ← “elderly”

IF AgeGroupTotals[0] >= AgeGroupTotals[1] AND AgeGroupTotals[0] >= AgeGroupTotals
[2] THEN
        BiggestAgeGroup ← “child”
ELSE IF AgeGroupTotals[1] >= AgeGroupTotals[0] AND AgeGroupTotals[1] >=
AgeGroupTotals [2] THEN
        BiggestAgeGroup ← “adult”
```

```
ELSE IF AgeGroupTotals[2] >= AgeGroupTotals[0] AND AgeGroupTotals[2] >=
AgeGroupTotals [1] THEN
        BiggestAgeGroup ← “elderly”
END IF
```

1 mark uses the AND Boolean operator/nested selection correctly
1 mark accesses values in array appropriately by index
1 mark finds biggest age group if child correctly
1 mark finds biggest age group if adult correctly
1 mark finds biggest age group if elderly correctly
1 mark all selection branches arrive at a return value and/or appropriate default value defined before selection.
1 mark returns the correct string value

Question 15 **(4 marks)**

John has never programmed an online game before and doesn’t quite know where to start. Luckily, he found some source code for a similar game on a website and plans to use it as a starting point. Briefly discuss one legal and one ethical issue associated with John using this code in his game.

2 marks for each relevant issue discussed
1 mark for identifying or making a superficial comment about a relevant issue.

Example responses

Copyright may have been breached, If John uses the code without first obtaining permission then he may be in breach of the Copyright Act of 1968 irrespective of how much he changes or modifies the code.

If John sells the game he produces he’s not only profiting off of someone else’s hard work but also denying someone else income that they’ve rightfully earned by putting out a product that directly competes with theirs despite not doing the work himself.

Accept all other relevant responses.

Refer to the following scenario to answer question 16
Violet is doing a project on global warming. She has found data online for the average global temperatures (the average of all temperatures measured around the world on any given day) recorded every day over the past year. Unfortunately, the temperatures are all in Fahrenheit and her project requires them to be in Celsius.

The project description provides the following formula for students to convert temperature readings from Fahrenheit to Celsius, where C is the reading in Celsius and F is the reading in Fahrenheit.

C = (F-32) x 5/9

Question 16 (21 marks)

Violet is trying to write a program that will:

- Read the daily Fahrenheit temperature readings from a file into an array.
- Convert all the Fahrenheit temperatures into Celsius and store the converted values in a new array.
- Output the average, maximum and minimum temperature values

So far, she has written the following pseudocode for the main module:

```
MODULE Main
        Fahrenheit ← FileToArray(datafile)
        Celsius ← []
        Call ConvertToCelsius(Fahrenheit, Celsius)
        Call CalculateTempStatistics(Celsius)
END Main
```

(a) The procedure ConvertToCelsius converts Fahrenheit temperatures in one array (provided through the first parameter) into corresponding Celsius temperatures in another array (provided through the second parameter). Assume both parameters are reference parameters. Write the pseudocode for the procedure in the space below.

(6 marks)

```
MODULE ConvertToCelsius(farray, carray)
        FOR i ← 0 TO farray.Length - 1
            carray[i] ← (farray[i] -32) * 5 / 9
        END FOR
END ConvertToCelsius
```

1 mark PROCEDURE or MODULE with correct name
1 mark parameters with appropriate names in correct order
1 mark FOR loop or equivalent test-first loop with counter
1 mark correct loop range. May use array.length -1 or 364. Other values may cause array index out of bound exception.
1 mark correctly uses loop variable or counter to address elements in each array
1 mark applies Fahrenheit to Celsius conversion formula correctly

-1 mark for missing END statements or other incorrect pseudocode conventions

(b) The procedure CalculateTempStatistics takes an array of temperatures as a parameter and outputs the average, maximum and minimum of all the temperatures. Write the pseudocode for the procedure in the space below. (15 marks)

```
MODULE CalculateTempStatistics(TempArray)
        Total ← 0
        Max ← TempArray[0]
        Min ← TempArray[0]

        FOR i ← 0 TO TempArray.Length – 1
                IF TempArray[i] > Max THEN
                        Max ← TempArray[i]
                END IF
                IF TempArray[i] < Min THEN
                        Min ← TempArray[i]
                END IF
                Total ← Total + TempArray[i]
        END FOR

        OUTPUT(Total / TempArray.Length)
        OUTPUT(Max)
        OUTPUT(Min)
END CalculateTempStatistics
```

1 mark initialises Total variable to 0
1 mark initialises Max variable to first array element or suitably large value (i.e. 1000)
1 mark initialises Min variable to first array element or suitably small value (i.e. -1000)
1 mark PROCEDURE or MODULE with correct name
1 mark correct parameter
1 mark FOR loop or equivalent test-first loop with counter. If array is iterated over more than once, still only award 1 mark.
1 mark correct loop range. May use array.Length -1 or 364. Other values may cause array index out of bound exception.
1 mark correctly uses loop variable or counter to address elements in array
1 mark selection to test each loop if current array element is greater than max
1 mark assigns current array element as new max
1 mark selection to test each loop if current array element is less than min
1 mark assigns current array element as new min
1 mark adds current array element to total each loop
1 mark correctly calculates average temperature
1 mark outputs average, max and min

-1 mark for missing END statements or other incorrect pseudocode conventions