



**Winter Examinations 2021/ 2022**

<b>Exam Code(s)</b>	3BCT1
<b>Exam(s)</b>	3rd Year Examination Computing Science and IT
<b>Module Code(s)</b>	CT331
<b>Module(s)</b>	Programming Paradigms
<b>Paper No.</b>	1
<b>Repeat Paper</b>	No
<b>External Examiner(s)</b>	Dr. Ramona Trestian
<b>Internal Examiner(s)</b>	Professor Michael Madden *Dr. Finlay Smith

**Instructions:** Answer any 3 questions. All questions will be marked equally. Each question is worth a maximum of 25 marks. The total (out of 75) will be converted to a percentage after marking.

**Duration** 2 hours  
**No. of Pages** 5  
**Discipline(s)** Computer Science  
**Course Co-ordinator(s)**

**Requirements:** None

**PTO**

1)

- a) How can you use the stack within 'C' code? Describe the steps involved – illustrate your answer with code snippets. What are the advantages and disadvantages of using the stack.  
(9 marks)
- b) In your own words describe the purpose of the following functions/operations in 'C'.
  - i) `sizeof()`
  - ii) `free()`(2 marks)
- c) What are the differences between `malloc` and `calloc()`.  
(2 marks)
- d) Write 'C' code that defines a structured type called `collegeStruct` with members that store the name of the College as a character array, the names of the Schools in the College as an array of pointers to strings and an array of integers that represents the number of students in each of the Schools. You can assume that there are 6 Schools in each College. Write a function called `deleteCollege` that accepts a pointer to a `collegeStruct` instance and frees all of the memory associated with the structure.  
(12 marks)

2)

- a) What are the advantages and disadvantages of using function pointers in 'C'? Illustrate your answer with code snippets.  
(5 marks)
- b) Write a generic sort function in 'C' that sorts an array of 20 elements. Your function should be passed a pointer to a print function that prints the elements of the type contained in the array (e.g. print an integer etc.) and then prints the elements in the array using that function. You should write a print function named `printInt`, and finally show you would call your generic function using this print function.  
(20 marks)

**PTO**

3)

a) Describe the features that distinguish Lisp from other programming languages. In particular what operations can be performed by Lisp?  
(3 marks)

b) In Lisp, what would the following function calls return?

- i) `(car (cdr (cdr '(1 2 3 4 5))))`
  - ii) `(cdr '((4 5 6 7)))`
  - iii) `(car (cdr (cdr '(1 (3 4) 5 6 7))))`
  - iv) `(car (cdr (car '((d e f) 1 2 3))))`
- (4 marks)

c) In Lisp, use `car` and `cdr` to return the following – in each case you can assume the lists have been stored in a variable called `lst`:

- i) Element `b` in the list `'(1 2 (b 1) 3 (4 5))`
  - ii) Element `2` in the list `'(8 ((7 4)) 2 1)`
  - iii) Element `2` in the list `'(9 (3 (4 2) 3) 0)`
  - iv) Element `'(1 2)` in the list `'((5 6) ( 1 (((1 2)))))`
- (4 marks)

d) Write a non tail recursive function in Scheme which takes 2 arguments (a list and a number) and returns a list of all of the numbers in the list that are either equal to the number or twice the number, eg if the number is 5 your list should contain all instances of 5 and 10. You can assume that each item in the list is a number and that there are no nested lists. For example, if the function is called `num_or_double`:

*(num\_or\_double '(2 4 6 8 1 3 5 7 4 3) 4) returns (4 8 4)*  
(6 marks)

e) Write a tail recursive version of your answer to part c). Make sure both your versions return lists with the elements in the same order.  
(8 marks)

**PTO**

4)

- a) Write a function in Lisp that takes one argument, a list of  $n$  elements. Your function should return the list repeated  $n$  times. Each copy of the list should have the first element removed. For example if the function is called *duplicate*:  
(*duplicate* '(1 2 3 4)) returns (1 2 3 4 2 3 4 3 4 4).  
(10 marks)

- b) Write a tail recursive function in Scheme that accepts a list of numbers and returns a list with every 3<sup>rd</sup> number of the original list removed. For example if the function is called *shorten\_list*:

(*shorten\_list* '(1 2 3 4 5 6 7 8 10)) returns (1 2 4 5 7 8)

(15 marks)

5)

- a) Give an example of facts, rules and queries all having the same name and arity. In your own words describe why this can be useful when writing Prolog code.  
(5 marks)
- b) In your own words describe the Closed World Assumption? How is the Closed World Assumption used in Prolog? What effect does it have on the facts that need to be provided to Prolog programs?  
(5 marks)
- c) Write a Prolog rule that finds the product of the largest and the smallest elements in a list. You can assume that all of the elements in the list are numbers greater than 0. For example:

?- *findProduct*([5, 9, 12, 4, 13, 22], *Res*).

*Res* = 88

as the smallest number is 4 and the largest number is 22

(15 marks)

**PTO**

6)

- a) In your own words describe how lists are handled in Prolog. Use examples to illustrate your answer, showing the operations you can perform on lists.  
(7 marks)

- b) Write code in Prolog that triples every 3<sup>rd</sup> element in a list, you can assume that all of the elements in the list are numbers. For example:

*?-triple\_list( [1,2,3,4,5,6,7], X).*

*X = [1,2,9,4,5,18,7]*

(8 marks)

- c) Write Prolog rules which take two lists as arguments. They should count the number of elements in the 2nd list that are less than the largest element in the 1<sup>st</sup> list. For example:

*?-countNumber([4,5,6,7,8,9,3] , [1,2,3,4,5,6,7,8,9,10], Count).*

*Count = 8*

As there are 8 numbers in the 2<sup>nd</sup> list that are less than 9 (the largest number in the 1<sup>st</sup> list).

(10 marks)