SEMESTER 1 EXAMINATIONS 2022/2023

MODULE:	CA320 - Computability and Complexity	
PROGRAMME(S):		
	CASE	BSc in Computer Applications (Sft.Eng.)
	ECSAO	Study Abroad (Engineering & Computing)
YEAR OF STUDY:	3, O	
EXAMINERS:	Dr. David Sinclair (Ext. 5510)	
TIME ALLOWED:	2 hours	
INSTRUCTIONS:	Answer 4 Questions. All questions carry equal marks.	

PLEASE DO NOT TURN OVER THIS PAGE UNTIL INSTRUCTED TO DO SO

The use of programmable or text storing calculators is expressly forbidden.
Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

Requirements for this paper (Please mark (X) as appropriate)

□ *Log Tables*
□ *Graph Paper*
□ *Dictionaries*
□ *Statistical Tables*

□ *Thermodynamic Tables*
□ *Actuarial Tables*
□ *MCQ Only - Do not publish*
□ *Attached Answer Sheet*

Section A

QUESTION 1 ***[Total marks: 25]***

1(a) [5 Marks]

Describe *list comprehension* in the Haskell Programming language using examples.

1(b) [8 Marks]

Write a Haskell function called `vowels` that take a string and *using list comprehension* returns a string containing all the vowels from the original string. Include the function type for `vowels`. For example, `vowels "Hello World"` would evaluate to `"eoo"`.

1(c) [12 Marks]

Without using the list concatenation operator ++, write a Haskell function that takes a list of lists as input and returns a list that is the concatenation of all the original lists. For example, if the input to the function was `[[3,2,1],[8,9],[1,2,3,4]]`, the function would evaluate to `[3,2,1,8,9,1,2,3,4]`.

[End Question 1]

QUESTION 2 ***[Total marks: 25]***

2(a) [5 Marks]

What are *higher order functions*?

2(b) [10 Marks]

Describe, with the use of examples, the higher-order Haskell functions map and filter.

2(c) [10 Marks]

Using the higher order functions `filter` and `map`, generate the sum of the list of all the squares of numbers from 1 to 100 that are even.

[End Question 2]

Section B

QUESTION 3 ***[Total marks: 25]***

3(a) [5 Marks]

In *computability* what does it mean when a problem is *undecidable*?

3(b) [13 Marks]

Prove that the *Halting Problem* is undecidable.

3(c) [7 Marks]

Describe how *reducibility* can be used to show a problem is undecidable.

[End Question 3]

QUESTION 4 ***[Total marks: 25]***

4(a) [5 Marks]

Formally describe the Big-O Notation. Rank five typical Big-O categories from "quickest" to "slowest".

4(b) [5 Marks]

In terms of Complexity, define NP-completeness.

4(c) [15 Marks]

The Cook-Levin Theorem states that *Satisfiability* is NP-complete. Give an outline of the proof of the Cook-Levin Theorem.

[End Question 4]

QUESTION 5 ***[Total marks: 25]***

5(a) [5 Marks]

In your own words briefly describe the difference between *computability* and *complexity*.

5(b) [20 Marks]

In the entrance to the School of Computing building there us a sign that say:

Redefining “Possible” with Computing

In the context of Computability, critique this phrase. Your answer should address the term *possible* in the context of Computability; and how could future technologies impact this phrase?

[End Question 5]

[END OF EXAM]