Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

SEMESTER 1 EXAMINATIONS 2021/2022

MODULE: CA320 - Computability and Complexity

PROGRAMME(S):

CASE	BSc in Computer Applications (Sft.Eng.)
ECSA	Study Abroad (Engineering & Computing)
ECSAO	Study Abroad (Engineering & Computing)

YEAR OF STUDY: 3

EXAMINERS: Dr. David Sinclair

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*	☐	*Thermodynamic Tables*
☐	*Graph Paper*	☐	*Actuarial Tables*
☐	*Dictionaries*	☐	*MCQ Only - Do not publish*
☐	*Statistical Tables*	☐	*Attached Answer Sheet*

Section A

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

1(a) [5 Marks]

What are *function types* in the Haskell programming language? How is a *function type* written in the Haskell programming language? What is the implication of the particular form the Haskell programming language uses for defining a *function type* and invoking functions?

1(b) [8 Marks]

In your own words describe *class constraints* in the Haskell programming language? Give an example of how to use a *class constraint*.

1(c) [12 Marks]

Write a Haskell function, **without using the built-in reverse function**, that tests whether or not a list is a palindrome, i.e. the list is equal to its reverse. The function should use class constraints to ensure that it will work for lists of integers, floats, and characters.

[End Question 1]

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

2(a) [5 Marks]

In the Haskell Programming language describe, **in your own words**, how *guards* operate with the use of example code.

2(b) [10 Marks]

Let a polynomial be represented as a list of coefficients. For example $a_3x^3 + a_2x^2 + a_1x^1 + a_0$ is represented as the list [a_0,a_1,a_2,a_3] . Define a Haskell function `evalPoly`, with the appropriate function type, which, given a polynomial and a value for x, will calculate the value of the polynomial for that value of x.

Hint: You may find the following identity helpful:
$a_nx^n + \ldots + a_2x^2 + a_1x + a_0 = a_0 + x(a_1 + x(a_2 + x(\ldots a_n)\ldots))$

2(c) [10 Marks]

Write a Haskell function `shortest`, with appropriate function type, that takes a list of lists and returns the shortest list in the list (and returns [ ] if the list of lists is empty).

[End Question 2]

Section B

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

3(a) [4 Marks]

Define the complexity class $\mathbf{P}$. $\underline{\textbf{In your own words}}$ describe what the class $\mathbf{P}$ represents.

3(b) [6 Marks]

Define the complexity class $\mathbf{NP}$. $\underline{\textbf{In your own words}}$ describe what the class $\mathbf{NP}$ represents. What is the fundamental difference between the classes $\mathbf{P}$ and $\mathbf{NP}$?

3(c) [15 Marks]

$\underline{\textbf{In your own words}}$ outline the structure of the proof of the Cook-Levin theorem. In particular, discuss the number of clauses generated in the proof.

[End Question 3]

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

4(a) [5 Marks]

Describe, $\underline{\textbf{in your own words}}$, the difference between a *deterministic finite automaton* and a *nondeterministic automaton*.

4(b) [10 Marks]

$\underline{\textbf{In your own words}}$ compare and contrast a *Linear Bounded Automaton*, a *Push-down Automaton* and a *Turing Machine*.

4(c) [10 Marks]

Design a *Turing Machine* that takes 2 unary numbers on the tape, where the first number is greater than the second number, and computes and writes a unary number to the tape that is the first number minus the second number.

[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 *p*ossible in the context of Computability; and how could future technologies impact this phrase?

[End Question 5]

[END OF EXAM]