



**Semester I Examinations 2019/2020**  
**PARTIAL SAMPLE PAPER (2) – NOT EMBARGOED**

<b>Exam Code(s)</b>	1MAO2, 1MAI1
<b>Exam(s)</b>	MSc in Computer Science (Artificial Intelligence), MSc in Computer Science (Artificial Intelligence) – Online
<b>Module Code(s)</b>	CT5132, CT5148
<b>Module(s)</b>	<b>Programming and Tools for Artificial Intelligence, Programming and Tools for Artificial Intelligence – Online</b>
 Paper No.	 1
 External Examiner	 Prof. Pier Luca Lanzi
 Internal Examiner(s)	 Prof. Michael Madden Dr. James McDermott *
 Programme Coordinator(s)	 Dr. Michael Schukat Dr. James McDermott

**Instructions**

**Answer ALL questions**

When writing code, comments and error-checking are not required except where explicitly stated.

<b>Duration</b>	2 Hours
<b>Number of pages</b>	3 (including this page)
<b>Discipline</b>	Computer Science

**Requirements**

Release in Exam Venue	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>
Release to Library	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>

## Question 2: Advanced Python

- (a) In what situations are *normalisation* and *canonicalisation* useful? Why? [5]
- (b) We have studied *time complexity*, that is how the time to run a piece of code will grow as the size of the input grows. Another interesting topic is *space complexity*, that is how the amount of memory required by a piece of code will grow as the size of the input grows. What do you think is the space complexity of the following code? [5]

```
def product_matrix(L):  
    """L is a list of numbers of length n"""  
    M = []  
    n = len(L)  
    for i in range(n):  
        M.append([])  
        for j in range(n):  
            M[i].append(L[i] * L[j])  
    return M
```

- (c) Explain and distinguish between all the uses of the `*` and `**` operators in Python. [5]
- (d) Explain and distinguish between the `@` symbol as used in (modern) Numpy and as used in plain Python. [5]

## Question 3: Data Science

- (a) In Scikit-Learn, what happens if we run `m.predict()` for a model `m` which has not yet been `fit()`ted? How does Scikit-Learn know whether it has been `fit()`ted? [5]
- (b) Suppose we have a Pandas DataFrame `d` with many columns. How can we extract from `d` a DataFrame with just the columns named `a` and `b`? How can we extract just column `a` as a 1D Numpy array? [5]
- (c) Give an example of a real-world application where we might see 5D data. [5]
- (d) What data format could we use to describe Santa's trajectory on Christmas Eve? What techniques could we use to extract useful features from this data, e.g. features that would be useful to distinguish his trajectory during city visits from his trajectory in the countryside? Assume that after leaving any house, he usually visits the nearest unvisited house next. [5]

## Question 4: Tools and Applications

- (a) Suppose we want to generate code. We could try choosing a random symbol, e.g. `for`, `while`, `True`, `0`, `<`, etc., one at a time, and then concatenating them. Will this work and why/why not? What alternatives are there? [5]
- (b) In the context of formal languages, define *language* and *sentence*. What does it mean to say that a string is *not a valid sentence*?
- (c) Describe an approach to programming a bot to play tic-tac-toe (noughts and crosses). It should attempt to learn from the other player's history. There is no need to provide complete code, but code snippets or pseudo-code may be used if needed. [5]