



Semester I Examinations 2019/2020
SAMPLE PAPER – NOT EMBARGOED

Exam Code(s)	1MAO2, 1MAI1
Exam(s)	MSc in Computer Science (Artificial Intelligence), MSc in Computer Science (Artificial Intelligence) – Online
Module Code(s)	CT5132, CT5148
Module(s)	Programming and Tools for Artificial Intelligence, Programming and Tools for Artificial Intelligence – Online
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.

Duration	2 Hours
Number of pages	4 (including this page)
Discipline	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 1: Basic Python

- (a) Re-implement the following code using `defaultdict` instead of `try-except`. Include any necessary import. [5]

```
d = {}
dna = "gattaca"
for s in dna:
    try:
        d[s] += 1
    except KeyError:
        d[s] = 1
```

- (b) Why will this code fail? How could it be fixed? [5]

```
d = {}
M = [[0, 1, 2],
      [3, 4, 5],
      [6, 7, 8]]
for L in M:
    d[L] = sum(L)
```

- (c) What is *duck typing* and how does it relate to *polymorphism*? [5]
- (d) Give a single floating-point value `x` which makes both of the following expressions `False`. [5]

```
x >= 3
x <= 3
```

- (e) Here is code which calculates the factorial function. Write a docstring containing three suitable doctests, one of which tests with an invalid input. [5]

```
def fact(n):
    if n < 0:
        raise ValueError("Bad input")
    if n == 1:
        return 1
    else:
        return n * fact(n - 1)
```

Question 2: Advanced Python

- (a) Suppose we have a *string* `s` containing an arithmetic expression in Python syntax, such as `s = "5 + 7"` or `s = "math.sqrt(12 / (1 + 1))"`. Write code which will calculate the value of that expression. Assume `math` has already been imported. [5]
- (b) The Fibonacci function below is very inefficient. Use *memoisation* to improve it: either implement memoisation by hand or `import` and use a decorator. [5]

```
def fib(n):
    if n in (0, 1):
        return n
    else:
        return fib(n - 1) + fib(n - 2)
```

- (c) What is *introspection*? Describe at least two Python facilities for introspection. [5]
- (d) Define the term *higher-order function* and give an example of one built-in to Python. [5]
- (e) What is the time complexity of this function with respect to the length n of the input `L`? How would it change if we rewrote this as a comprehension? [5]

```
def f(L):
    result = []
    for x in L:
        for y in L:
            result.append(x * y)
    return sum(result)
```

Question 3: Data Science

- (a) “A Pandas `DataFrame` is like a dictionary with ____ as keys and ____ as values.” Fill in the blanks. [5]
- (b) What is the difference between `np.save` and `np.savetxt`? Which will result in a larger file, on disk? [5]
- (c) In a scientific context, what is the motivation and purpose of code like this? [5]

```
import random
random.seed(0)
```

- (d) Demonstrate in code the core workflow for Scikit-Learn linear regression, including `import`, training, evaluation, inspection of the parameters of the trained model, to prediction on unlabelled data. The code should use datasets `X_train`, `X_test`, `y_train`, `y_test` and an array of query points `X_query`. You may assume they have already been created in the right format. [5]
- (e) Describe the use of inheritance and *mixins* in the Scikit-Learn API. [5]

Question 4: Tools and Applications

- (a) Suppose we have an *electrocardiogram* (ECG) time-series signal, stored in a Numpy array. Suppose we pass this array to a Fourier transform function. What format will the result be, and how can we interpret it to produce an estimate of the heart rate? [5]
- (b) “NetworkX represents graphs using a *dict-of-dicts-of-dicts*”. Explain this statement with the help of a small example. [5]
- (c) Rewrite the following undirected graph in adjacency matrix format. What is the main disadvantage of this format relative to the NetworkX format? [5]
 $G = (V, E)$ where $V = \{0, 1, 2, 3, 4\}$ and $E = \{(0, 1), (0, 2), (0, 3), (2, 4), (3, 4)\}$.
- (d) Consider an automated taxi. It has three states: *wandering*, *going to pick-up*, *going to drop-off*. Its sensors can deliver these three inputs: *pick-up order received*, *pick-up complete*, *drop-off complete*. Draw a *finite state machine* suitable for controlling this taxi. [5]
- (e) Write down a grammar in BNF format which can generate any of the three “sentences” below (and others) but cannot generate the last. Remember to say which symbol is the start symbol. [5]

```
the cat sat on the mat
cat sat on the mat
the mat sat on mat

hat the cat the mat
```