

Computational Management Science 1

Fall 2020 Final

registration number:
(Do not write your name on the test - just the 7 digit student id number.)

All examples are evaluated using the Python programming language, version 3.8.

1. (6 points) Writing Code

(a) (3 points, ≤ 5 minutes) Functions

Write a function that takes exactly one non-negative integer as argument and returns the factorial of that number. Use either a loop or recursion to compute the solution.
 $\text{factorial}(0) = 1$, $\text{factorial}(5) = 5 * 4 * 3 * 2 * 1 = 120$.

(b) (3 points, ≤ 5 minutes) Classes and data structures

Implement a simple `Student` data structure in Python. The data structure must be capable of storing the data of a student in this course: `first_name`, `last_name`, `id_number`, `scores`. The first three are passed when constructing an instance while `scores` must be initialized to an empty list.

Write a **minimalistic** class (`__init__()`). You don't need to implement any functionality, just a class that stores the required data. Note that your implementation must be self-contained and must not rely on any third party functionality not provided by you. Don't forget to write docstrings in order to receive full points.

2. (6 points, ≤ 10 minutes) Relational Database Design

Design a relational database to store the following information:

We want to store a list of machines for which we know the name and the speed. In addition, we want to store a list of jobs for which we know a name, release time, deadline and size (integer).

- (a) Every job is assigned to a unique machine on which it has to be produced.
In addition, please also give an example on how to store the case of 2 machines and 3 jobs, the first two being scheduled on the first machine, the third scheduled on the 2nd machine.
- (b) Every job can be assigned on none, one or multiple machines.
In addition, please also give an example with two machines and two jobs. The first job should be assigned to the 2nd machine and the 2nd job should be assigned to both machines.

3. (9 points, ≤ 10 minutes) Libraries

(a) (1 points)

What is the purpose of a named tuple? Which library provides this data structure?

(b) (3 points)

What is the most important category of automated tests. Is their execution generally fast or slow? When should they be executed?

(c) (3 points)

What is the purpose of doctests? Provide one example of a function with a doctest (use valid syntax).

(d) Finally, name one thing you like about this course and one thing that should be improved in the future (be honest!) (2p).

4. (12 points, ≤10 minutes) Reading and Understanding Code

What is the output of the following code snippets? Write exactly what the output of each snippet is if the snippet is the sole content of a Python file. If the output is an error message, it is enough to write "ERROR". If there is no output, write "-"

(a) Loop

```
n = 10
d = 6
work = True
while work:
    if n / d == 0:
        print(d)
        work = False
    d -= 1
```

(b) Simple calculation

```
a = 2
b = 3
c = a * b**2 + 2 * b - 1
print(c)
```

(c) Function

```
def funnyswap(a, b):
    c = a
    a = b
    b = c
    return a, b

a = 5
b = 10
c, d = funnyswap(a, b)
print(a, b, c, d)
```

(d) Lists and tuples

```
values = [-5, 4, 3]
values[0] = 5
```

(e) String operations

```
s = "Everyone wants to get the grade 1 in CMS."
print(s[3])
print(s[:5] + ' ' + s[-4:-1])
```

(f) List comprehension

```
integers = [2*i for i in range(5)]
print(max(integers))
```

5. (9 points, ≤ 10 minutes)

(a) (3 points)

What is a zero-sum game? In addition to the general definition, give one example of a zero-sum game and one example of a game that is not a zero-sum game.

(b) (3 points)

What is recursion? What alternative programming concept can you always use instead of recursion? Write a short code example of a recursive function.

(c) (3 points)

What is polymorphism? Give one example where and how you would use polymorphism.

6. (6 points, ≤5 minutes) Student Activities

A fictitious university offers a very large number of different spare time activities. Each student is allowed to enroll in exactly one activity or may choose not to enroll in any. The data is currently stored in a Python dictionary which uses student ids as keys and the name of the student's activity as value. If a student picked no activity, None is used. In order to know who is enrolled in a given activity, the dictionary should be transformed into another dictionary with activities as keys and a list of enrolled students as values. The sort order of students in the lists is not relevant. Write a function `transform(before)` that returns a new dictionary in the desired format. For example

```
before = {'9825756': 'archery',
          '0315423': 'table tennis',
          '1303451': 'archery',
          '0834154': None}
after = {'archery': ['9825756', '1303451'],
         'table tennis': ['0315423'],
         None: ['0834154']}
```