

Computational Management Science 1

Spring 2019 Final

registration number: SAMPLE SOLUTION

(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 3.6.

1. (6 points) Writing Code

(a) (3 points, ≤5 minutes) Functions

Write a function `sum_if(iterable, predicate)` in Python that takes a sequences of numbers length and a predicate (boolean valued function). It must return the sum of each element if the element fulfills the predicate. e.g.

`sum_if((1, 3, 4, 7, 8), lambda x: x % 2) ⇒ 11`

`sum_if([3, 5, 1, 2, 4], lambda x: x > 3) ⇒ 9`

Add a proper docstring to receive full points.

```
def sum_if(iterable, predicate):
    """Return the sum of each element in iterable fulfilling 'predicate'."""
    total = 0
    for e in iterable:
        if predicate(e):
            total += e
    return total
```

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

Implement a simple data structure in Python. The data structure must be capable of storing a circle in a two dimensional drawing application (i.e. the coordinates of the center as well as the radius). Write a **minimalistic** class (`__init__(.)`). You don't need to implement any functionality, just a class that stores the required data. Don't forget to write docstrings in order to receive full points.

```
class Circle(object):
    """A circle in a two-dimensional Euclidean plane."""

    def __init__(self, x: float, y: float, radius: float):
        self.x = x
        self.y = y
        self.radius = radius
```

2. (6 points, ≤10 minutes) Correct Mistakes

The following code contains 6 syntax errors/ typos. Clearly mark and correct the mistakes. (hint: you don't need to understand what the function does to correct the mistakes as there are no logical errors; assume that all required classes are available → just look for syntax errors)

```
class Point(object):
    """A Point in a two-dimensional Euclidean plane."""

    def __init__(self, x: float, y: float):
        self.__x = x
        self.__y = y

    def __str__(self):
        return (f'Point(x={self.x}, y={self.y})')

    @property
    def x(self):
        return self.__x

    @property
    def y(self):
        return self.__y
```

3. (9 points, ≤10 minutes) 3rd Party Libraries

(a) (2 points)

What is the purpose of argparse?

It is a library for parsing command-line arguments.

(b) (3 points)

What is a Series? What is a DataFrame? Which 3rd party package provides them?

A Series is a one-dimensional array with axis labels.

A DataFrame is a two-dimensional data structure with labeled rows and columns.

Both are part of the pandas package by Wes McKinney.

(c) Name (in total) two advantages and / or disadvantages of PuLP over other modeling languages (2p).

+ full power of a general purpose programming language for pre- and post-processing as well as heuristics

+ pulp uses a permissive open source license (can be used free of charge even in commercial environments)

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

UP TO THE STUDENTS...

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) Simple calculation

```
num = 3
num++
```

ERROR

(b) Loop

```
prices = (4, 5, 2)
total = 0.0
for price in prices:
    total += price
print("Total:", total)
```

Total: 11.0

(c) Function

```
def add(a, b):
    return a + b
perimeter_right_triangle(5, 3)
```

ERROR

(d) List

```
l = (5, 3, 2)
l.append(1)
print(sum(l))
```

ERROR

(e) Numpy

```
import numpy as np
m = np.zeros(4).reshape(2,2)
print(m[0,1])
```

0.0

(f) Lists

```
l = [2, 4, 3]
print(l.append(1))
```

None

5. (9 points, ≤10 minutes) Various

(a) (3 points)

What is a generator expression (PEP 289)? Which key advantage does it have? Provide at least one example.

A high performance, memory efficient generalization of list comprehensions.

```
sum(x*x for x in range(10))
```

(b) (3 points)

What is inheritance? What is composition? Which relationship types does each of them model?

In object-oriented programming, inheritance is the mechanism of basing a class upon another class.

Object composition is a way to combine objects or data types into more complex ones.

Aggregation is very similar but does not imply ownership.

Inheritance models an “is-a” relationship while composition and aggregation model a “has-a” relationship.

(c) (3 points)

What is git? What can be done with it? What are its limitations and downsides?

Git is a distributed version-control system for tracking changes in text files. For example, it allows for collaborative features like pull (merge) requests. It is also possible to find the commit responsible for a regression via `git bisect`. Like other version control systems, git does not offer its full set of features when working with binary files.

6. (6 points, ≤5 minutes) Writing Files

In one of your Python programs, a series of key-value pairs are the central results. Your employer asks you to design and implement a function that stores permanently these results. Explain how you approach the problem and why you decide to do so. Implement the function `save(filename: str, pairs: dict) -> None` that takes a filename string and the key value pairs. The function should write the key-value pairs to a file with the given name. Any representation of the data is ok as long as it allows to easily read the data back in from the file (but you don't need to implement the reading function, just the writing function). Don't forget to document the function in order to receive full points.

I would approach a solution as simple as possible. Since key-value pairs are used in many structured text file types like json and yaml I would resort to one of them. Since Python offers built-in support for json that I would prefer using that module.

```
import json

def save(filename: str, pairs: dict) -> None:
    """Save the given results to a JSON file."""
    with open(filename, 'w', encoding='utf-8') as f:
        json.dump(pairs, f)
```