Matriculation number:

Practice Exam: Python for Physicists

1) Give a short description of the types of python containers we have discussed: tuples, lists, sets, and dictionaries. When might you want to use a list instead of a tuple? (5 P)

Solution:
List - a mutable, or changeable, ordered sequence of elements
Tuple - unchangeable list (cannot be changed)
Set - contains only unique values (no duplicates)
Dictionary - a structure that contains pair of keys and values corresponding to the keys

List instead of tuple when the algorithm requires modification of the list (like adding element, deleting element)

2) Explain the difference between a for loop and a while loop. When might a while loop be preferable? (4 P)

Solution:
For loop iterates through the elements in the object (list, dictionary)
While loop checks a certain condition after each iteration

If the number of iterations is not known but the condition must be met, a while loop is preferable.

3) Write down a generic function definition and explain the different components. (4 P)

Solution:
def <function_name>([<parameters>]):

    <statement(s)>

Function takes arguments as input, process them via statements in the function body and returns result as output (but may also just process statements with no return)

4) What are the possible return types for a python function? (4 P)

Solution:
It can return any possible type

5) What is the difference between a global and a local variable? (4 P)

Solution:

-Local variables live only inside of functions and are not accessible from outside.

-Global variables are accessible both inside and outside of functions.

6) Label the local and global variables for this code snippet. What do you expect for the print statements for this code snippet? (5 P)

```
def add_one(count):
        count += 1          # local
        return count        # local

count = 1                   #global
print(count)                #global

add_one(count)              # global
print(count)                # global

count = add_one(count)      #global
print(count)                #global
```

Output: (1,1,2)

7) Identify the problem in this function definition. (3 P)

```
def is_detectable(luminosity, threshold=1e-11, distance):
        flux = calc_flux(luminosity, distance)
        return flux > threshold
```

Solution: the default argument has to be in the end

8) What do the two syntaxes below mean; what type of argument does the function take? (4 P)

a) `def myfunction(*args)`

b) `def myfunction(**kwargs)`

Solution:
*args: allow to pass any number of arguments to the fuction

**kwargs: any number of arguments with values are passed, i.e. keyword arguments (e.g. grade=10)

9) What is the output of the following code snippet? (3 P)

```
(lambda x: x**2)(2)
```

Solution: 4

10) Write down a generic class definition and explain the different components. (6 P)

```
class Physicist:                              #class definition

    def __init__(self, name, xp, money):     # initializer method

        self.name = name                      # instance attribute
        self.xp = xp
        self.money = money
```

11) When should you use a class? When can you simply use a container or a function? (4 P)

Classes combine data and functionality. If you are only interested in storing data, or only interested in a particular functionality, you don't need a class. You can stick to containers for data storage, and to normal functions for implementing a behavior.

12) Label the class attribute and the instance attribute in this code snippet. (4 P)

```
import math

class Circle:
       num_instances = 0              # class attribute
       def __init__(self, radius=1):
       self.radius = radius     # instance attribute
       Circle.num_instances += 1
```

13) Explain the role of setter and getter methods in a class. (4 P)

Getter method: gets the value of an attribute
Setter method: sets the value of an attribute

14) What is the output after executing the following lines in python? (4 P)

a)
```
my_list = [1, 2, 3, 4, 5]
my_list[::-1]
```
Output: [5,4,3,2,1]

b)
```
a = '1234'
a[-1]
```
Output: '4'

15) When might you need to use pickle for file writing/reading, rather than e.g. JSON? When is it preferable to use JSON, and why? (2 P)

Pickle is python specific, you need to know what's inside the file. It is very useful for a quick serialization of Python objects.
JSON is ideal for data interchange between different systems and languages.

JSON is human-readable dictionary-like structure
pickle is a binary file (not human readable)

16) What are the advantages of using numpy arrays, rather than native python iterables coupled with for loops? (3 P)

-faster computations, less memory usage, array operations take less lines of codes, simpler structure

17) What is vectorization? How does it relate to numpy arrays? (4 P)

Vectorization: Going from operating with single value at a time to an array of values at a time.

All numpy arrays support vectorization.

18) Will this code snippet run successfully? If not, why not? (3 P)

```
import numpy as np

matrix1 = np.arange(10).reshape(2,5)
vector1 = np.arange(5)
matrix1 > vector2
```

Solution: vector2 is not defined

19) Write a pseudo-code for a recursive function that calculates the Golden Ratio (the quotient between successive pairs of Fibonacci numbers). (10P)

```
FUNCTION GoldenRatio(i, fibo_number=False):

    IF fibo_number IS TRUE:

        IF i == 0:
            RETURN 0

        ELSE IF i == 1:
            RETURN 1

        ELSE:
            RETURN GoldenRatio(i-1, TRUE) + GoldenRatio(i-2, TRUE)

    ELSE:

        number1 = GoldenRatio(i-1, TRUE)  # Fibonacci(i-1)

        number2 = GoldenRatio(i-2, TRUE)  # Fibonacci(i-2)

        dividendo = number1 + number2      # Fibonacci(i)

        divisore = number1                 # Fibonacci(i-1)

        RETURN dividendo / divisore
```

20) Write a pseudo-code for generating a spectrum following the form N~A*E^-1.5 for a set of energies E linearly spaced between 0.1 and 5, and fitting the spectrum with a functional form. (10 P)

```
GENERATE A = constant

CREATE ARRAY E = 0.1 to 5, linearly spaced

CREATE ARRAY N = empty array, same size as E

#Generate spectrum:

N[i] = A * E[i]^-(1.5)

DEFINE FITTING_FUNCTION(A_fit, E):

    RETURN A_fit * E[i]^-(1.5)

# Use FITTING_FUNCTION on generated spectrum
```

21) Write a pseudo-code for a function that generates a random password with at least 8 digits, including at least one letter, one integer, and one special character.

```
DEFINE FUNCTION password():
    SET password = empty list
    numbers = [1,2,3,4,5,6,7,8,9,0]
    letters = ['a', ... ,'z']
    spec_char = ['$','%','&','/','(',')','=','?']

    APPEND random(numbers) TO password
    APPEND random(letters) TO password
    APPEND random(spec_char) TO password

    TOTAL = numbers+letters+spec_char

    FOR i in RANGE(8-3)
        APPEND random(TOTAL) TO password
    SHUFFLE password
    RETURN password
```