Code : BFVM19PRO	**Exam:** Assessment Programming	
Date: 15-09-2021	**Time:** 08:30 – 10:00	**School:** ILST
Room: H186/H188a	**Class:** DSLSR	**Duration:** 90’
Teacher: WERD **Lecturer can be reached during exam: 06-14680604.**		**Number of pages: 5**
Resources: Cheatsheet test_in.fa		**Additional resources:** None
Hand in exam: No **Hand in scrap paper: No**		
Particular details: Documentation is made available on your Desktop through links or readable documents		
Name student **Class:**		**Studentnumber:**

- Do not copy paste from this document! Any command or example needed, should be typed by you.

- You can use any Integrated Development Environment you are familiar with. Or even a text editor if that works for you. Tools available are: Pycharm, Geany and Idle.

- Your grade will be calculated as:
  - *1 + (points scored/maximum score * 9)*

- At the end of the exam you can just logout of your computer and leave the files on your Desktop.

HERE STARTS THE ACTUAL TEST

Create a Python file with the name: *yourname.py.*
The Python file should be used to answer all questions.

Question 1 (9 points)

Q1_1: Create a string variable called *q1_1* that will hold the following text: I will rock this exam! (3 points).

Q1_2: Create a dictionary variable called *q1_2* that will hold the combination of first name and last name of students. For this you can use mock students (3 points).

Q1_3: Given the next problem what datatype would you use. Explain in clear wording why you choose this datatype (3 points).

> Problem: Program needs to store many different measured student heights in cm.
> Save your answer in a variable called *q1_3*

Question 2 (21 points)

Q2_1: Create separate functions for every question in this test (7 points).

Q2_2: Create a function *to_be_imported* inside a Python module called *sub_module.py* and import it in your script. The function should determine the length of a word given as an argument and return it. Catch the length of the word and print it in your main script (7 points).

Q2_3: Create a function that will read a filename given as argument. The file should be read line by line. Remove line enters (newlines). Skip the header lines. A header line is a line starting with **>**.

For example: >1

Finally write the non header lines (the lines containing only the letters ‘A’, ‘T’, ‘C’, ‘G’, to a new file called *q3_output.txt*. Test your function using the filename ‘*test_in.fa’* as argument. Use *with(open)* for reading and writing (7 points).

Question 3 (5 points)

Create your own exception class *Not4* inheriting from the base Exception class. Iterate the following numbers: 1, 2, 3, 5, 6. Throw the exception *Not4* on **NOT** finding the number 4 Catch this error and print: 4 not found when this error is catched or: Answer to everything found otherwise (5 points)

Question 4 (20 points)

Q4_1: rewrite your *yourname.py* file so that all functions are called from a central main function in the *yourname.py* script.

Add a check to test if this is the main running script, and only when this is the main script call the main function (5 points).

Q4_2: Add Python documentation in the correct format on the module level and also for every function you've created. Add some inline comments to show that you know how to write these (5 points).

Q4_3: Debug the following piece of code (10 points).

```
def fixed:
    import: sys
    print sys.vesion
    x = 'this correct??"
    y
    for i in x;
        y =+ i


    if length(y) * 3 = 42
        outcome == "Great succes!"
    else
        outcome == "bummer"


    return outcome
```

Question 5 (20 points)

Q5_1: Create a class called *Course* that can be used to store the following information (5 points):

- Course name
- Course lecturer
- Course exam form, can be one of: assignment or exam
- holds a list with student names attending the course

Q5_2: When the class is created, it should be mandatory to be created with the Course name and lecturer. Keep track of the number of created classes, save this in a class variable *called number_of_courses* (5 points)

Q5_3: Create the following methods (5 points):

- get number of students in class
- add student(s) names to a list, if there are less than 15 students in the class, if there are more than 15 print: class is full!

- set the exam form, check if it is one of assignment or exam and then assign this to the class if the exam_form is not allowed, show this by printing form not allowed

Q5_4: Finally, show that your class can be used by (5 points):

Creating three courses: python, datascience and omics, if you don't know who teaches, just use a fake name here.

Print the number of total Course classes created by printing the class variable of one the courses.

Try to set the exam form of omics to practical

Add a bunch (minimal 2) students to a course and lastly print the number of students in that course

Question 6 (10 points)

Q6: Use list comprehension to calculate the inverse of a number (1/number) and then calculates that number to the third power. Do this for each number in the following list: 1, 2, 3, 4, 5, 6. Store the results in a tuple, use round to round the inverse before you save it in the tuple (10 points)

Expected result: [(1, 1), (0, 8), (0, 27), (0, 64), (0, 125), (0, 216)]

END