

G.A.U. Faculty of Engineering
CEN488 Python Programming

(Duration: 90 minutes)

Midterm Examination

April 15, 2022

Dr. Kamil Dimililer

No:	
Name:	
Signature:	

1. Write a Python program where the user enters positive real numbers from the keyboard. When the user enters the first non-positive number, the number entry process ends, and the program displays the number and the average of the entered numbers.

Solution 1:

```
number_list=[]
while True:
    number=float(input("Enter a positive number: "))
    if number<1:
        break
    number_list.append(number)
print(len(number_list),
      "numbers are entered\nThe average of the numbers is {:.2.4f}".format(
        sum(number_list)/len(number_list)))

#The following code was enough for the midterm examination
#print(len(number_list),"numbers are entered\nThe average of the numbers is ",
#      sum(number_list)/len(number_list))
```

Solution 2:

```
num=0
sum_of_numbers=0.0
while True:
    number=float(input("Enter a positive number: "))
    if number<1:
        break
    num+=1
    sum_of_numbers+=number
print(num,"numbers are entered\nThe average of the numbers is {:.2.4f}".format(sum_of_numbers/num))

#The following code was enough for the midterm examination
#print(num,"numbers are entered\nThe average of the numbers is",sum_of_numbers/num)
```

2. Write a complete Python script to sketch the 2-D graph (plot) of $f(x)$ for $x=[0,6]$, where the following expression defines the $f(x)$ function:

$$f(x) = 1 + \sum_{n=1}^{40} (-1)^{n-1} \frac{4}{(2n-1)\pi} \cos((2n-1)\pi x)$$

Solution 1:

```
|from numpy import linspace
|from math import pi,cos
|from matplotlib.pyplot import plot,show

y_axis=[]
t_axis=linspace(0,6)
for x in t_axis:
    tot=1.0
    for n in range(1,41):
        tot+=4.0*((-1)**(n-1))*cos((2*n-1)*pi*x)/(2*n-1)/pi
    y_axis.append(tot)
plot(t_axis,y_axis)
show()
```

Solution 2:

```
from numpy import linspace
from math import pi,cos
from matplotlib.pyplot import plot,show,xlabel,ylabel

y_axis=[]
x_axis=linspace(0,6,200)
for x in x_axis:
    tot=1.0
    for n in range(1,41):
        tot+=4.0*((-1)**(n-1))*cos((2*n-1)*pi*x)/(2*n-1)/pi
    y_axis.append(tot)
plot(x_axis,y_axis)
xlabel("x")
ylabel("y")
show()
```

3. Write a Python function named "pattern" that accepts two arguments (an integer and a character) and prints the character to form a triangular shape, as shown below:

With 6 and '*'

```
  *
 * *
* * *
* * * *
* * * * *
* * * * * *
```

With 5 and '8'

```
  8
 8 8
8 8 8
8 8 8 8
8 8 8 8 8
```

With 7 and 'K'

```
  K
 K K
K K K
K K K K
K K K K K
K K K K K K
K K K K K K K
```

Solution:

```
def pattern(N, S):
    '''N is an integer and
    S is expected to be only one character'''
    for row in range(N):
        for col in range(N-row-1):
            print(end=" ")
        for col in range(row+1):
            print(S, end=" ")
        print("")
```

4. Write a Python function that accepts lists, where each list contains "name_surname" and CGPA of a student as its arguments. The number of students is not known. The function prints only the student's name with the highest CGPA, and it returns the list of that student. If more than one student achieved the highest grade, multiple names should be printed, and the function should return multiple lists.

Solution:

```
def student_with_highest_cgpa(*students):  
    '''Finds and prints students with the highest grade  
        Each student ["student name",CGPA]'''  
    student_highest=[]  
    student_highest.append(["",0.0])  
    for student in students:  
        if student[1]>student_highest[0][1]:  
            student_highest.clear()  
            student_highest.append(student)  
        elif student[1]==student_highest[0][1]:  
            student_highest.append(student)  
    print("The student(s) with the highest CGPA: ")  
    for st in student_highest:  
        print(st[0])  
    return(student_highest)
```

5. Determine the output of the following code:

```
grade_pts={'A':4.00,'A-':3.7,'B+':3.30,'B':3.00,'B-':2.70,
           'C+':2.3,'C':2.0,'C-':1.7,'D+':1.3,'D':1.0,'F':0.0}

class Student:
    def __init__(self,name):
        self.name=name
        self.courses=[]
        self.cgpa=0.0

    def add_a_course(self,course,grade):
        self.courses.append([course,grade])

    def calc_gpa(self):
        total_credit=0.0
        total_points=0.0
        for crs in self.courses:
            total_credit+=crs[0][1]
            total_points+=crs[0][1]*grade_pts[crs[1]]
        self.cgpa=total_points/total_credit

course1=["MT111",4]
course2=["PS111",3]
course3=["ENG101",3]
course4=["ENG103",3]

student1=Student("Ahmet Arkin")
student2=Student("John Seed")
student1.add_a_course(course1,'A-')
student1.add_a_course(course2,'B')
student1.calc_gpa()
student1.add_a_course(course3,'C')
student2.add_a_course(course1,'D')
student2.add_a_course(course3,'C-')
student2.add_a_course(course4,'B+')
student2.calc_gpa()
print(student1.name,student1.cgpa)
for n in student1.courses:
    print("\t",n[0][0],n[1])
print(student2.name,student2.cgpa)
for n in student2.courses:
    print("\t",n[0][0],n[1])
```

Output:

```
Ahmet Arkin 3.4
        MT111 A-
        PS111 B
        ENG101 C
John Seed 1.9
        MT111 D
        ENG101 C-
        ENG103 B+
```

6. Write a python function (name it "my_swap") that accepts a list as the argument and returns a new list that contains the same elements. The new list consists of elements taken from the beginning and end of the argument, respectively: For example, the first and last elements of the argument list are the first and the second element of the new list; the second element from the beginning and the end are the third and fourth elements of the new list and so on.

Solution:

```
def my_swap(lst):  
    '''Swaps the elements of the list lst'''  
    ret=[]  
    N=len(lst)  
    for n in range(N//2):  
        ret.extend([lst[n],lst[-(n+1)]])  
    if N%2==1:  
        ret.append(lst[N//2])  
    return ret
```