



Laboratory Exercise 1

Python Programming for DSP

Submitted by:
Group #8

SO	PI	Category	Exceptional 4	Acceptable 3	Marginal 2	Unacceptable 1	Score
b	1	Compliance 30%	All procedures were followed, the output is as expected, additional related functionalities were augmented.	All procedures were followed and the output is as expected.	All procedures were followed but the output is not as expected.	Did not follow the set procedures	
b	1	Analysis 20%	Data interpretation is professionally written with appropriate and clear illustrations.	Data is clearly and correctly explained with proper illustrations.	Data is not clearly explained, has minor flaws, or no illustrations	There is no data explanation about the data or results.	
b	1	Validity 20%	The implementation uses the concepts and principles of the experiment as well as advanced topics.	The implementation uses the concepts and principles of the theory discussed for the experiment.	Implementation did not clearly express the use of theory discussed for the experiment.	There is no implementation.	
b	1	Interpretation 20%	The conclusion is professionally written and points the theories in the experiment and its implications in engineering.	The conclusion points to the main ideas and applications of the theory in the experiment.	The conclusion does not point out the main ideas and applications of the theory in the experiment.	There is no conclusion.	
		Format and Clarity 10%	Follows the prescribed format, observes proper and technical grammar, and observes proper citation and referencing according to IEEE journal standards.	Follows the prescribed format, observes proper and technical grammar, and observes proper IEEE citation and referencing.	Did not follow the prescribed format, has poor grammar, or incorrect citations and references scheme.	Did not follow the prescribed format, has poor grammar, and has no citations and references.	
		TOTAL SCORE					

Group Members			
STUDENT NUMBER	NAME	CONTRIBUTION	SCORE
202011184	Castro, Billy Boy Chino	Document	
201912082	Chipongian, John Patrick	Program	
202150578	David, Dwight Matthew G	Document	

Submitted to:
Engr. Dexter James L. Cuaresma

Date:
08/19/2024

OBJECTIVES

- This laboratory exercise aims to help students in reviewing their python programming for linear algebra and DSP.
- To use Python, Anaconda, Jupyter Notebook as programming environment.

DISCUSSION

Welcome to Python Fundamentals

In this module, we are going to establish or review our skills in Python programming. In this notebook we are going to cover:

- Variables and Data Types
- Operations
- Input and Output Operations
- Logic Control
- Iterables
- Functions

Variable and Data Types

```
x = 1
a,b = 0, -1
```

```
type(x)
```

```
y = 1.0
type(y)
```

```
x = float(x)
type(x)
```

```
s,t,u = "0", '1', 'one'
type(s)
```

```
s_int = int(s)
s_int
```

Operations

Arithmetic

```
a,b,c,d = 2.0, -0.5, 0, -32

### Addition
S = a+b
S

### Subtraction
D = b-d
D

### Multiplication
P = a*d
P

### Division
Q = c/a
Q

### Floor Division
Fq = a//b
Fq

### Exponentiation
E = a**b
E

### Modulo
mod = d%a
mod
```

Assignment Operations

```
G, H, J, K = 0, 100, 2, 2

G += a
G

H -= d

J *= 2
J

K **= 2
K
```

Comparators

```
res_1, res_2, res_3 = 1, 2.0, "1"
true_val = 1.0

## Equality
res_1 == true_val

## Non-equality
res_2 != true_val

## Inequality
t1 = res_1 > res_2
t2 = res_1 < res_2/2
t3 = res_1 >= res_2/2
t4 = res_1 <= res_2
t1
```

Logical

```
res_1 == true_val
```

```
res_1 is true_val
```

```
res_1 is not true_val
```

```
p, q = True, False
conj = p and q
conj
```

```
p, q = True, False
disj = p or q
disj
```

```
p, q = True, False
nand = not(p and q)
nand
```

```
p, q = True, False
xor = (not p and q) or (p and not q)
xor
```

I/O

```
print("Hello World")
```

```
cnt = 1
```

```
string = "Hello World"
print(string, ", Current run count is:", cnt)
cnt += 1
```

```
print(f"{string}, Current count is: {cnt}")
```

```
sem_grade = 82.243564657461234
name = ""
print("Hello {}, your semestral grade is: {}".format(name, sem_grade))
```

```
w_pg, w_mg, w_fg = 0.3, 0.3, 0.4
print("The weights of your semestral grades are:\n\t{:.2%} for Prelims\n\t{:.2%} for Midterms, and\n\t{:.2%} for Finals.".format(w_pg, w_mg, w_fg))
```

```
x = input("enter a number: ")
x
```

```
name = input("Kimi no nawa: ")
pg = input("Enter prelim grade: ")
mg = input("Enter midterm grade: ")
fg = input("Enter finals grade: ")
sem_grade = None
print("Hello {}, your semestral grade is: {}".format(name, sem_grade))
```

Looping Statements

While

```
## while Loops
i, j = 0, 10
while(i<=j):
    print(f"{i}\t|\t{j}")
    i+=1
```

For

```
# for(int i=0; i<10; i++){
# printf(i)
# }

i=0
for i in range(10):
    print(i)
```

```
playlist = []
print('Now Playing:\n')
for song in playlist:
    print(song)
```

Flow Control

Condition Statements

```
numeral1, numeral2 = 12, 12
if(numeral1 == numeral2):
    print("Yey")
elif(numeral1>numeral2):
    print("Hoho")
else:
    print("Aww")
print("Hip hip")
```

Functions

```
# void DeleteUser(int userid){
#     delete(userid);
# }

def delete_user (userid):
    print("Successfully deleted user: {}".format(userid))

def delete_all_users ():
    print("Successfully deleted all users")
```

```
userid = 0
delete_user(0)
delete_all_users()
```

```
def add(addend1, addend2):  
    return addend1 + addend2  
  
def power_of_base2(exponent):  
    return 2**exponent
```

Lambda Functions

```
x = 4
```

```
def f(x):  
    return 2*(x*x)-1  
f(x)
```

```
g = lambda x: 2*(x*x)-1  
print(g(x))
```

MATERIALS

Software:

- Python
- Anaconda
- Jupyter Notebook

PROCEDURES

```
'''  
Create a grade calculator that computes for the semestral grade of a course.  
Students could type their names, the name of the course, then their prelim,  
midterm, and final grade.  
The program should print the semestral grade in 2 decimal points and should  
display the following emojis depending on the situation:  
happy - when grade is greater than 70.00  
laughing - when grade is exactly 70.00  
sad - when grade is below 70.00  
'''  
happy, lol, sad = "\U0001F600","\U0001F606","\U0001F62D"
```

At the same, the user must have the option to show their equivalent grades based on the given image below.

SECTION 3. POLICY ON GRADING SYSTEM

3.1 The University adopts the numerical grading system with the following points and their corresponding description for Undergraduate Programs and Graduate Programs:

Semestral Grade	Grade Point Equivalent	Letter Equivalent	Description
100	1.00	A	Excellent
99			
98			
97			
96			
95	1.25	A-	Superior
94			
93			
92			
91			
90	1.50	B+	
89			
88			
87			
86			
85	1.75	B	Average
84			
83			
82			
81			
80	2.00	B-	
79			
78			
77			
76			
75	2.25	C+	Passed
74			
73			
72			
71			
70	2.50	C	
<70			
	2.75	C-	Failed
	3.00	D	
	5.00	F	DROPPED
	6.00	DR	
	0.00	NG	NO GRADE

Save your works as Lab1_GroupNo. And upload it to the submission file.

RESULTS AND DISCUSSION

The code we made is designed to calculate Student's semester grade based on their prelim, midterm and final grade, and then the program will display the grade in the desired format, it's either Standard Grade (SG) or Grade Point Equivalent (GPE). The program also contain emoji that reflects the student's performance. When the program has been run it collects first the input from the user, icluding what's the student's name, course and grades for prelim, midterm and final grade. After inputting prelim, midterm and final grade it will ask how they want their grade to be displayed. Either as a numerical Standard Grade (SG) or Gradepoint Equivalent (GPE).

The computation of semester grade is calculated using a general weighted average formula: $sem_grade = (prelim_grade * 0.3) + (midterm_grade * 0.3) + (final_grade * 0.4)$. Depending on the user's choice, the program will print the grade in chosen format. If "GPE" is selected, the program converts the numerical grade to a Grade Point Equivalent and print it with an emoji: 😊 for grades between 1.00 and 2.75, 😐 for a grade of 3.00, and 😞 for grades below 3.00. But, if "SG" is selected, the numerical grade is displayed, accompanied by an emoji: 😊 for grades between 71 and 100, 😐 for a grade of 70, and 😞 for grades below 70. This program also include error handling , when the user enter unrecognized option by the code it will print or display "Invalid Input!".

CONCLUSION

In conclusion, This Laboratory Activity effectively review us the python programming with practical application of operations, assignment operations, logical, I/O, condition statements, functions, and looping statements. By utilizing Google Colab as an alternative to Anaconda and Jupyter Notebook. We are able to work on it despite tool constraints. The activity culminated in the creation of a Python program designed to calculate the final grade and display the student's semestral grade base on their grade in prelim, midterm and final grades. The program offers 2 types of display it's either displayed in Standard Grade (SG) or Grade Point Equivalent (GPE) format, and it also includes emojis to visually represent the student grade. We were also able to implement the error handling to manage invalid input. Overall, this exercise not only reinforced programming skills but also demonstrated the importance of adaptability and maximizing available alternative solutions for the problem u encounter whether it's on application or software programming.