

PYTHON INTERNSHIP

Name: Kartik V Naik

RegisterdEmail: [4nm18cs075@nmamit.in](mailto:4nm18cs075@nmamit.in)

PersonalEmail: [kartiknaikhanehalli@gmail.com](mailto:kartiknaikhanehalli@gmail.com)

Submitted to,

Mr.Sobin Sunny

Mr.Arun

|  |  |
| --- | --- |
| S.no | Contents |
| 1 | Introduction |
| 2 | About Dlithe |
| 3 | Problem 1 |
| 4 | Problem 2 |
| 5 | Problem 3 |
| 6 | Problem 4 |
| 7 | Problem 5 |

INDEX

INTRODUCTION

**Python Programming**

We were made aware of the extent of possibilities through Python programming and how it has made its mark in becoming the most demanding programming language out there through the extensive library offered by its community and endless possibilities of Machine Learning and Artificial Intelligence through its libraries.

**Internship Overview**

At the beginning of the course, we were provided with information about its history and how it became the most popular language we know of now and then went on with the introduction to some of the flavour it gives for its community and the overwhelming extent of library it has to offer. This course went from the basics to being able to solve intermediate-level coding questions.

We were also given sufficient coding questions regularly by our mentor for each concept being taught to implement the concepts on the go, which included coding questions on patterns and also questions from google kickstart. We were also been guided to write short and time-efficient codes to come up with creative logic for the assignment questions.

**Concepts**

Our Mentor made us understand the whole concept by defining its syntax, functionality, and effective usability of the same using some examples.

* Identifiers and their properties.
* Different types of operators and their functions.
* Different types of data types starting of with Numbers and ending with frozen set. We were taught about its functionalities with different operators.
* We were been given in-depth knowledge of the syntax of input and output statements and various parameters inside the statements.
* Conditional Control Structures and its functions
* Functions of break and continue statements used in loops and how it can be implemented to write efficient code.
* Even though Python doesn't come with a ternary operator, we were taught how to modify the code to achieve it.
* Concepts of functions and methods.
* Functions and methods of various data types and an in-depth tour of the same using some examples.
* User-defined functions and types of arguments.
* We were also given some information about some inbuilt libraries of python.

**Conclusion**

This Journey has been fun and at the same time knowledgeable. This course has helped me gain fundamental knowledge about python and on the whole, has taught me how to go forward with solving any kind of programming problems implementing my logic expressed with Python.

ABOUT DLITHE

DLithe is EduTech company located in Bengaluru, serving corporate and academic institutions. The company vision is to build an agile workforce towards Industry needs.

The company has developed software solutions for a client in Healthcare domain. The company also provides technical consultancy in upgrading the platforms (legacy to modern), resource onboarding, competency development services.

The company has 20+ technical consultants in delivering various services of the company.

The company is lead by Arun Rajpurohit and Sridhar Murthy, industry professionals with a rich experience in IT services and product development.

PROBLEMS

**Problem statement 1**: A dam reservoir's capacity is 250e100 cubic meter where it holds 199.5e85 cu.m water. A rain occurred in the area which gave 25e8 cu.m water but 25% amount of this rain water ran-off useless. The remaining water from this rainfall flowed to the reservoir and there it was collected. On another day, a heavy storm increased the water level of this dam to its 15% of the current water level. Ground water sources contributed 5% to reservoir's current level. 5% of the present level of reservoir evaporated and later 15% amount of water was passed for irrigation to arid regions.  
  
**Task**  
Write a python program using less number of lines of code to find the current water level of the reservoir using the given data?

**Solution:**

|  |
| --- |
| #current amount of water in reserviour  c=**199.5e85**  #after first rainfall(wastage of25% of 25e8)  r=**0.75**\***25e8**  #total amount of water in reserviour  res=c+r  #15% of current amount of water from rain  res+=res\***0.15**  #5% from ground resource  res+=res\***0.05**  #15% for irrigation  res-=res\***0.15**  #5% evoporated  res-=res\***0.05**  **print**("current water level is",res,"cu.m") |

**Problem statement 2:** A receptionist has the duty of noting the details of visitors walking-in to the office. She notes in the register their name, phone number, place coming from and body temperature. In a view to change this pen-paper task to a computer task, use your python programming skills to make a solution for this operation.  
  
The program should be able to gather the four inputs mentioned earlier. Besides, a database/text file should be created in the local device (computer) with this data. Bring out the best solution! When there are no more entries, any solution as per programmer's choice may be used to terminate the execution of the program.

**Solution:**

|  |
| --- |
| **import** **datetime**  x = datetime.datetime.now()  dt=x.strftime("%d")+"\_"+x.strftime("%b")+"\_"+x.strftime("%Y")  **while** (1):  e = input("Do you want to enter entry: y/n")  **if**(e=='y'):  name=input("enter the name")  phno=int(input("enter the phone nummber"))  place=input("enter place")  temp=float(input("enter temperature"))  list = [('Name: ', name),('PhNo: ',phno),('place: ',place),('Temperature: ',temp),('Date\_time: ', datetime.datetime.now())]  **with** open(dt+".txt", "a") **as** f:  f.write("-----------------------------**\n**")  f.write('**\n**'.join('{} {}'.format(x[0],x[1]) **for** x **in** list))  f.write("**\n**-----------------------------**\n**")  f.close()  **else**:  **break** |

**Problem statement 3:** You might have heard about Cyptography, haven't you? Sending confidential messages with clean encoding and decoding. It has great importance especially in cyber intelligence as well as defence ministry. There are many kinds of encyphering techniques prevailing now.   
  
Have you hear about Caesar's encypher?  
This is an ancient and simple encypher method.   
Look at this encoded message  : L ORYH BRX   
When decoded, it looks this way : I LOVE YOU   
  
Wondering how?  
In the previous encoding we used an alphabet right shift of 3. It means every alphabet will be considered with third corresponding alphabet when encoded.   
  
Finally we will have,  
  
Shift value: 3   
Actual   : A B C D E F G ...... U V W X Y Z   
Encoded : C D E F G H I ........ X Y Z A B C   
  
So, now **write a Python program for Caesar's encypher** that can accept any input and shift value from user and perform the encypher operation. Nevertheless, proceed with decypher operation if some encoded message is given as input

**Solution:**

|  |
| --- |
| **def** Encrpytingchar(char):  **if**(char!=" "):  **return** chr((((ord(char)+3)+65)%26)+65)  **return** " "    **def** CaesorEncrypter(a):  encrypted\_msg=""  **for** i **in** range(0,len(a)):  encrypted\_msg+=Encrpytingchar(a[i])  **return** encrypted\_msg  **def** Decrpytingchar(char):  **if**(char!=" "):  **return** chr((((ord(char)-3)+65)%26)+65)  **return** " "    **def** CypherDycrypter(a):  decrypted\_msg=""  **for** i **in** range(0,len(a)):  decrypted\_msg+=Decrpytingchar(a[i])  **return** decrypted\_msg  msg=input("Enter the Message").upper()  Encrpyted\_msg=CaesorEncrypter(msg)  print(Encrpyted\_msg)  Decrypted\_msg=CypherDycrypter(Encrpyted\_msg)  print(Decrypted\_msg) |

**Problem statement 4:** Get an input from the user and generate pattern in the order as that can be referred below:  
NB: Concept of nested loops will be helpful  
  
  
INPUT:   
5  
OUTPUT:  
\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

**Solution:**

|  |
| --- |
| **def** halfPyramidStar(N):  **for** i **in** range(N):  **for** j **in** range(0, i + 1):  print("\*", end = "")  print()    N=int(input("Enter input"));  halfPyramidStar(N); |

**Problem statement 5:** Get an input from the user and generate pattern in the order as that can be referred below:  
NB: Concept of nested loops will be helpful

INPUT:  
3  
OUTPUT:  
\*  
\*\*  
\*\*\*  
\*\*  
\*

**Solution:**

|  |
| --- |
| **def** halfDiamondStar(N):  **for** i **in** range(N):  **for** j **in** range(0, i + 1):  print("\*", end = "")  print()    **for** i **in** range(1, N):  **for** j **in** range(i, N):  print("\*", end = "")  print()    N=int(input("Enter input"));  halfDiamondStar(N); |

THANK YOU