...OLVING THROUGH PYTHON LAB

Instruction	: 3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Credits	: 1.5	Semester End Examination	:	70 Marks
		Semester End Exam Duration	:	3 Hours

Course Objectives

1. To understand the syntax of Python and learn language basics such as strings, operators, decision making statements and data structures.
2. To inculcate modular programming using modules and packages.
3. To deal with file I/O related problems.
4. To introduce web application development and database connectivity in Python.
5. To explore data visualization tools.

Exercises:

1. a. Write a Python program to create all possible strings by using 'a', 'e', 'i', 'o', 'u
   b. Write a Python program to create all possible permutations from a given collection of distinct numbers.
   c. Write a Python program to check the priority of the four operators (+, -, *, /).
2. a. Write a Python program that accepts a sequence of lines (blank line to terminate) as input and prints the lines as output (all characters in lower case).
   b. Write a Python program to check the validity of password input by users.
   - At least 1 number between [0-9].
   - At least 1 character from [$#@].
   - Minimum length 6 characters.
   - Maximum length 16 characters.
   - At least 1 letter between [a-z] and 1 letter between [A-Z].
3. a. Write a Python function to check whether a number is in a given range.
   b. Write a Python function that prints out the first n rows of Pascal's triangle.
4. Write a Python program to make a chain of function decorators (bold, italic, underline etc.) in Python.
5. a. Write a Python program to find the list of words that are longer than n from a given list of words.
   b. Write a Python program to create a list by concatenating a given list which range goes from 1 to n.
   c. Write a Python program to find missing and additional values in two lists.
6. Write a Python program to sort the list of elements using the following sorting techniques:
   a) Merge Sort
   b) Quick Sort
7. a. Write a Python program to sort a dictionary by key.
   b. Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary.

Scanned with CamScanner

c. Write a Python program to create a dictionary from two lists without losing duplicate values.

8. a. Write a Python program to remove empty tuple(s) from a list of tuples.

   b. Write a Python program to solve the Knapsack problem Using Greedy Method.

9. Write a Python program to unzip a list of tuples into individual lists.

10. a. Write A Program to Create a Class which Performs Basic Calculator Operations

    b. Write A Program to Create a Class in which One Method Accepts a String from the User and Another Prints it.

11. Write a Program to find the minimum cost spanning tree using

    a) Prim's Algorithm

    b) Kruskal's Algorithm

12. Write a program to find Single Source Shortest path.

13. Write A Program that Reads a text file and counts the number of occurrences of a given word.

14. Write A Program that Reads a CSV File, replace the blank cells with null operations and group the data with their statistical means.

15. Build a Python framework for student database using Django.

16. a. Write a Python program to slice ndarray with in the given range.

    b. Write a Python Program to create a surface plot and mesh plot using Matplot lib.

17. Familiarize yourself with following Python libraries: NumPy, pandas, matpoltlib.

    a) Read quick NumPy tutorial.

    b) Read quick pandas tutorial.

    c) Read quick matplotlib tutorial.

18. Build the linear regression model for the given dataset using Pandas and Numpy

19. Write a program to implement K Nearest Neighbor Algorithm (Use 75% of the data as the training set, fix the random state as 0 and the k value as 2)

20. Write a program to predict the class of testing sample using Baye's classification.

21. Implement K- Means clustering on IRIS dataset.

Note: Programs from 1 to 14 are mandatory and Programs from 15 to 21 are optional.

Course Outcomes: At the end of the course, the student will be able to

CO 1 : Write Python programs using control statements and data structures.
CO 2 : Develop modular programming skills using modules and packages.
CO 3 : Develop applications to read and write data from/to files and explore the Object-Oriented concepts.
CO 4 : Familiar with Django fr...

Scanned with CamScanner

31/03/2021

Lab programs

(1)

(a) Write a python program to create all possible strings by using 'a', 'e', 'i', 'o', 'u'.

Ans)

```
from itertools import permutations
ch = ['a', 'e', 'i', 'o', 'u']
string = permutations (ch)
print (string)

for str in string :
        print ("".join (str))
```

*output:-

```
a e i o u
a e i o u
a e o i u
a e o u i
a e u i o
.
.
.
.
.
```

Scanned with CamScanner

(1) (b) Write a python program to check all possible permutations from a given collection of distinct members.

Ans)

```
from itertools import permutations
dist_num = ['1', '3', '5', '7']
n = permutations(dist_num)

for num in n:
    print("".join(num))
```

*output :-

```
1 3 5 7
1 3 7 5
1 5 3 7
1 5 7 3
1 7 3 5
1 7 5 3
3 1 5 7
3 1 7 5
3 5 1 7
3 5 7 1
3 7 1 5
3 7 5 1
5 1 3 7
5 1 7 3
5 3 1 7
5 3 7 1
5 7 1 3
5 7 3 1
7 1 3 5
7 1 5 3
7 3 1 5
7 3 5 1
7 5 1 3
7 5 3 1
```

Scanned with CamScanner

(2) (c) Write a python program to check the priority of the four operators (+, -, *, /)

Ans)

```
prior = {
        '+' : 0
        '-' : 0
        '*' : 1
        '/' : 1
    }

def check_priority (op1, op2) :
        if prior[op1] >= prior[op2] :
            return op1 + " has higher priority than " + op2
        else
            return op2 + " has less priority than " + op2
print ( check_priority ('+', '/') )
print ( check_priority ( '/', '+') )
print ( check_priority ( '-', '/') )
print ( check_priority ('*', '-') )
```

*output :

```
+ has less priority than /
/ has higher priority than +
- has less priority than /
* has higher priority than -
```

Scanned with CamScanner

(2) (a) Write a python program that accepts a sequence of lines & blank lines to terminate as input & print the lines as output (all characters in lower case)

Ans)

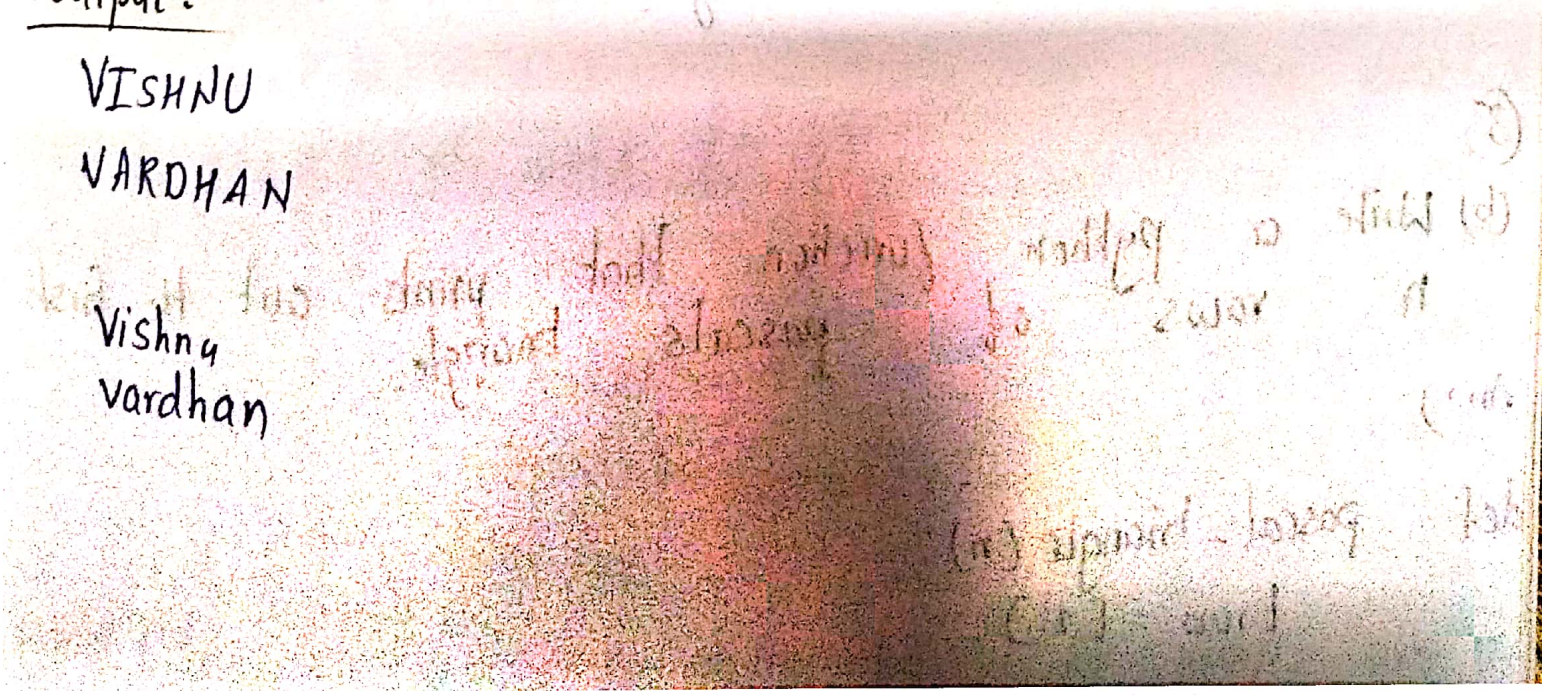
```
ch = [ ]
  n = input ()
if n :
    ch.append (n.lower())
else
    break;

for c in ch:
    print (c)
```

*output :-

```
VISHNU
VARDHAN

Vishnu
vardhan
```



Scanned with CamScanner

③ (a) Write a python function to check whether a number is in given range.

Ans)

```
def test_range (n):
    if n in range (5,10):
        print ("%s is in the range.", %str(n))
    else:
        print ("%s is outside the range.", %str(n))
```

*~~output~~:

```
test_range (6)
test_range (12)
```

*output:

6 is in the range.
12 is outside the range.

③ (b) Write a python function that prints out the first n rows of pascals triangle.

Ans)

```
def pascal_triangle (n):
    trow = [1]
    y = [0]
    for x in range (max(n,0)):
        print (trow)
        trow = [l+r for l,r in zip (trow+y, y+trow)]
    return n>=1
```

Scanned with CamScanner

pascal-triangle (6)

+output:-

[1]

[1, 1]

[1, 2, 1]

[1, 3, 3, 1]

[1, 4, 6, 4, 1]

[1, 5, 10, 10, 5, 1]

5)

(a) Write a python program to find the list of words that are longer than n from a given list of words.

Ans)

```
def long_words (n, str):
    word_len = []
    txt = str.split (" ")
    for x in txt:
        if len(x) > n :
            word_len.append (x)
    return word_len
print (long_words (3, "The quick brown for jumps over
                    the lazy dog"))
```

+output:

['quick', 'brown', 'jumps', 'over', 'lazy']

Scanned with CamScanner

(5)

(b) Write a python program to create a list by concatenating a given list which range goes from 1 to n.

Ans)

```
my-list = ['p', 'q']
n=4
new-list = [ '{} {}'.format(x,y) for y in range (1,n+1)
             for x in my-list ]
print ( new-list)
```

*output:

['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4']

(5)

(c) write a python program to find missing & additional values in two lists.

Ans)

```
list1 = ['a', 'b', 'c', 'd', 'e', 'f']
list2 = ['d', 'e', 'f', 'g', 'h']
print ("Missing values in second list : ", ','.join (set (list1).
                                        difference (list2)))
print ("Additional values in second list : ", ','.join(set(list2)
                                        difference (list1)))
```

Scanned with CamScanner

*output :-

Missing Values in Second list : c, b, q
Additional Values in Second list : g, h

(Q7)

(a) Write a python program to sort a dictionary by key.

Ans)

```
dict = {
        'a' : 'apple',
        'b' : 'bat',
        'c' : 'cat'
        'e' : 'ear',
        'd' : 'dog'
    }

for key in sorted(dict):
    print("%s : %s" %(key, dict[key]))
```

*output:

a : apple
b : bat
c : cat
d : dog
e : ear

Scanned with CamScanner