Python Programming

Practical Record

Lab Manual

SRI CHAITANYA
TECHNICAL CAMPUS
COLLEGE OF ENGINEERING & TECHNOLOGY
COLLEGE OF BUSINESS MANAGEMENT
(Approved by AICTE, NEW DELHI & Affiliated to JNTU, Hyderabad)

www.srichaitanyaengg.com
E-mail : director8a.sctc@gmail.com

Sheriguda (V), Ibrahimpatnam (M), R.R. Dist. - 501 510 - A.P.
Ph : 08414 - 223222, 223223 Fax : 08414 - 222678

MCA I Yr. -I Semester

SRI CHAITANYA
TECHNICAL CAMPUS

COLLEGE OF ENGINEERING & TECHNOLOGY
COLLEGE OF BUSINESS MANAGEMENT

(Approved by AICTE, NEW DELHI & Affiliated to JNTU, Hyderabad)

Sheriguda (V), Ibrahimpatnam (M), R.R. Dist. - 501 510 - A.P.

CERTIFICATE

This is to certify that Mr / Ms _____ has satisfactorily completed experiments in Python Programming laboratory as prescribed by Jawaharlal Nehru Technological University, Hyderabad.

Department Master of Computer Applications Roll No_____

Branch MCA Academic Year 2025-2026

INTERNAL EXAMINER HEAD OF THE DEPT.

EXTERNAL EXAMINER PRINCIPAL

INDEX

Sl.No.	Date	Name of the Experiment	Page No.	Remarks

PYTHON PROGRAMMING LAB

MCA I Year I Sem.

L	T	P	C
0	0	2	1

Course Objectives:

- To install and run the Python interpreter
- To learn control structures.
- To Understand Lists, Dictionaries in python
- To Handle Strings and Files in Python

Course Outcomes: After completion of the course, the student should be able to

- Develop the application specific codes using python.
- Understand Strings, Lists, Tuples and Dictionaries in Python
- Verify programs using modular approach, file I/O, Python standard library
- Implement Digital Systems using Python

Note: The lab experiments will be like the following experiment examples

Week -1:

1. i) Use a web browser to go to the Python website http://python.org. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
   ii) Start the Python interpreter and type help() to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3. i) Write a program to calculate compound interest when principal, rate and number of periods are given.
   ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read name, address, email and phone number of a person through keyboard and print the details.

Week - 2:

1. Print the below triangle using for loop.

   5
   4 4
   3 3 3
   2 2 2 2
   1 1 1 1 1
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

Week - 3:

1. i) Write a program to convert a list and tuple into arrays.
   ii) Write a program to find common values between two arrays.
2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.
3. Write a function called palindrome that takes a string argument and returnsTrue if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.

Week - 4:

1. Write a function called is_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.
2. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.
   i). Write a function called remove_duplicates that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
   ii). The wordlist I provided, words.txt, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.
   iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
   ii) Remove the given word in all the places in a string?

iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?

4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5:

1. i) Write a python program that defines a matrix and prints
   ii) Write a python program to perform addition of two square matrices
   iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
3. Use the structure of exception handling all general purpose exceptions.

Week-6:

1. a. Write a function called draw_rectangle that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
   b. Add an attribute named color to your Rectangle objects and modify draw_rectangle so that it uses the color attribute as the fill color.
   c. Write a function called draw_point that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
   d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called draw_circle that draws circles on the canvas.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 7

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. Write a function that reads a file *file1* and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCES:

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
3. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
4. Think Python, Allen Downey, Green Tea Press
5. Core Python Programming, W. Chun, Pearson
6. Introduction to Python, Kenneth A. Lambert, Cengage

DEPARTMENT OF Masters of Computer Application

Vision & Mission

Vision

* To achieve high quality in technical education that provides the skills and attitude to adapt to the global needs of the Information Technology sector, through academic and research excellence.

Mission

* To equip the students with the cognizance for problem solving and to improve the teaching learning pedagogy by using innovative techniques.

* To strengthen the knowledge base of the faculty and students with motivation towards possession of effective academic skills and relevant research experience.

* To promote the necessary moral and ethical values among the engineers, for the betterment of the society.

Quality Policy

* Strives to inculcate the students with the world class Technical Knowledge, Entrepreneurial Competence and Social Ethics by providing continual improvement and innovation in the curriculum; based upon well-defined measurements and best practices.

* Develop faculty competencies, creativity, empowerment and accountability through faculty development programs and show strong management involvement and commitment.

INDEX Python Programming	Page No
Week -1:	
1. i) Use a web browser to go to the Python website http://python.org. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.	
ii) Start the Python interpreter and type help() to start the online help utility.	
2. Start a Python interpreter and use it as a Calculator.	
3. i) Write a program to calculate compound interest when principal, rate and number of periods are given.	
ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points	
4. Read name, address, email and phone number of a person through keyboard and print the details.	
Week - 2:	
1. Print the below triangle using for loop. 5 4 4 3 3 3 2 2 2 2 1 1 1 1 1	
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)	
3. Python Program to Print the Fibonacci sequence using while loop	
4. Python program to print all prime numbers in a given interval (use break)	
Week - 3:	
1. i) Write a program to convert a list and tuple into arrays.	
ii) Write a program to find common values between two arrays.	
2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.	
3. Write a function called palindrome that takes a string argument and returnsTrue if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.	
Week - 4:	
1. Write a function called is_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.	
2. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.	
i). Write a function called remove_duplicates that takes a list and returns a new list with only the unique	
elements from the original. Hint: they don't have to be in the same order.	
ii). The wordlist I provided, words.txt, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.	
iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.	
3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'	
ii) Remove the given word in all the places in a string?	

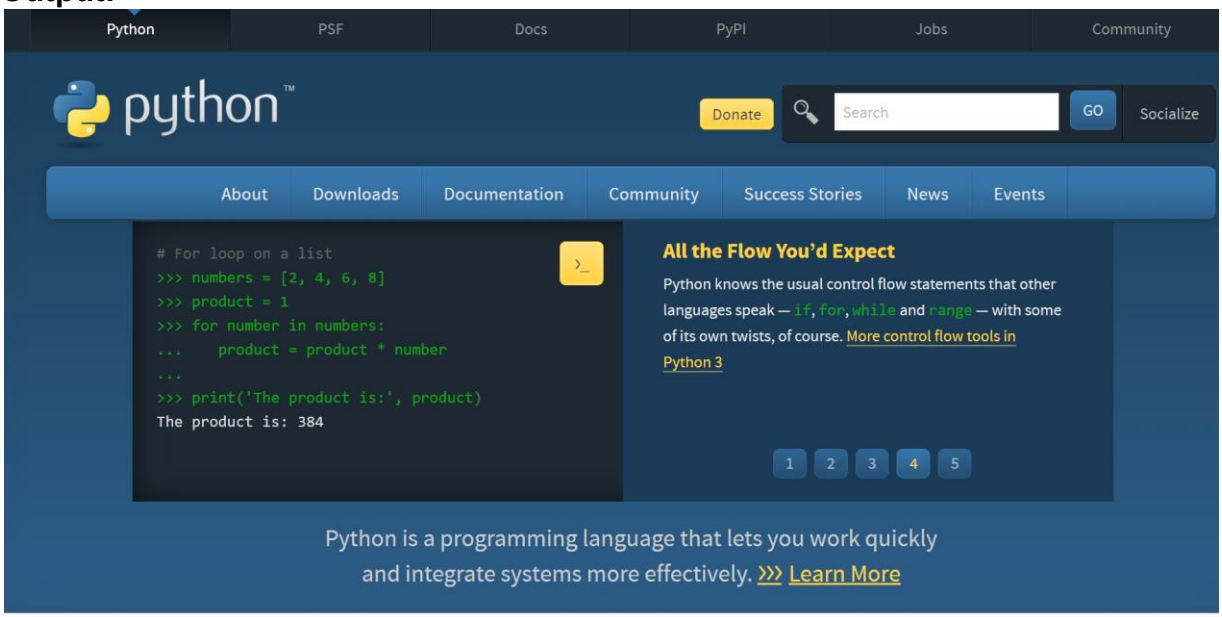
iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?	
4. Writes a recursive function that generates all binary strings of n-bit length	
Week - 5:	
1. i) Write a python program that defines a matrix and prints	
ii) Write a python program to perform addition of two square matrices	
iii) Write a python program to perform multiplication of two square matrices	
2. How do you make a module? Give an example of construction of a module using different geometrical shapes	
and operations on them as its functions.	
3. Use the structure of exception handling all general purpose exceptions.	
Week-6:	
1. a. Write a function called draw_rectangle that takes a Canvas and a Rectangle as arguments and draws a	
representation of the Rectangle on the Canvas.	
b. Add an attribute named color to your Rectangle objects and modify draw_rectangle so that it uses the color	
attribute as the fill color.	
c. Write a function called draw_point that takes a Canvas and a Point as arguments and draws a representation	
of the Point on the Canvas.	
d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called draw_circle that draws circles on the canvas.	
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.	
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.	
Week- 7	
1. Write a Python code to merge two given file contents into a third file.	
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.	
3. Write a Python code to Read text from a text file, find the word with most number of occurrences	
4. Write a function that reads a file file1 and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.	
Week - 8:	
1. Import numpy, Plotpy and Scipy and explore their functionalities.	
2. a) Install NumPy package with pip and explore it.	
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR	
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder	
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.	

Week -1:

1. i) Use a web browser to go to the Python website http://python.org. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
Program:

Step 1: Enter in web browser **www.http://python.org**

Output:



Step 2: click on **Doc** in main menu
Output:

Step 3: Enter Topic **Quick Search** , click on **Go**
Example enter math()

Output:

math (Python module, in math — Mathematical functions)

cmath (Python module, in cmath — Mathematical functions for complex numbers)
math — Mathematical functions

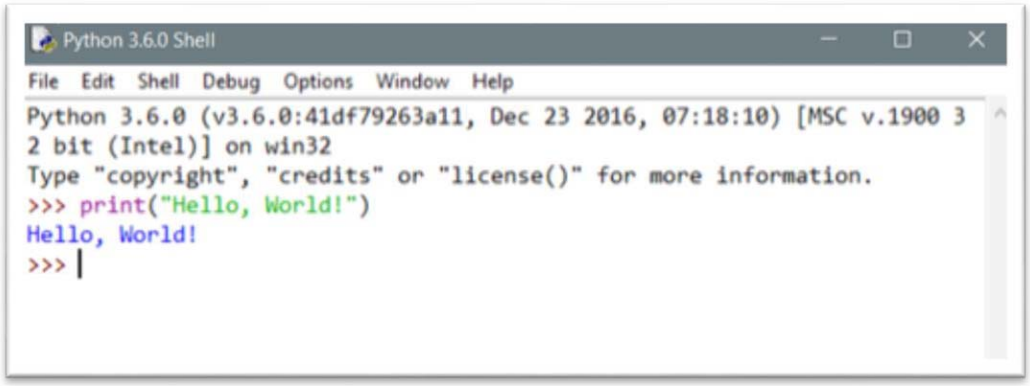
Step 4: click on math()

Output:
math.gcd(integers*):** Return the greatest common divisor of the specified integer arguments.
math.fmod(x, y): Return fmod(x, y), as defined by the platform C library.
math.fabs(x): Return the absolute value of x.
math.factorial(n): Return n factorial as an integer.
math.ceil(x): Return the ceiling of x, the small integer greater than or equal to x.

Continue Searching From **Ste3** To **Step** 4 To Show Related Information As Document:

1.ii) Start the Python interpreter and type help() to start the online help utility.

Step 1: click on start button and click on python 3.9
Output:



Step 2: help(arguments) function : help() function is a built-in function takes arguments as classes, functions, keywords to provides information about them.

Step 3: type **help**(function_name or “keyword_name” or operator_name or extra)

Example
>>>help(print)
Output:

```
print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

Step 4:
>>> import keyword
>>> help("keywords")
Output:
list of the Python keywords. Enter any keyword to get more help.

False break for not None class from or True continue global pass
__peg_parser__ def if raise and del import return as elif in try assert else is while async except lambda with await finally nonlocal yield

Step 5: >>> help("int")
Output:

Help on class int in module builtins:

class int(object)
| int([x]) -> integer
| int(x, base=10) -> integer
| Convert a number or string to an integer, or return 0 if no arguments
| are given. If x is a number, return x.__int__(). For floating point
| numbers, this truncates towards zero.

Continue **step** 5 to get helpful information of each one related to python by this help command

2. Start a Python interpreter and use it as a Calculator.
Program:

Python interpreter used as Calculator. The Python REPL shows three arrow symbols >>> followed by a blinking cursor. Programmers type commands at the >>> prompt then hit [ENTER] to see the results according to BODMAS rule priority of operators**.** To do following.

- **Arithmetic**: - ****, //, %, / , *, +, -**
- **Statistics** : - mean, median, mode, stdev.
- **Exponents and Logarithms:** log, log10, exp, e, pow(x,y), sqrt
- **Trigonometry**: sine, cosine, and tangent

Arithmetic: - ****, //, %, / , *, +, - are follows** BODMAS rule priority.

```
>>> 23-4*2
        15
>>> 45+6**2
        81
>>> 5000*(1+4/100)**2-5000
        408.0000000000009
>>> 66//7
      9.00
>>> 85%9+(3**2+5//3)-15/3
```

```
    9.0
```

Statistics: (mean(), median(), mode(), stdev()) By importing statistics , these functions are used

```
>>> from statistics import mean, median, mode, stdev
>>> Data = [60, 83, 83, 91, 100]
>>> mean(Data)
   83.4
>>> median(Data)
   83
>>> mode(Data)
   83
 >>> stdev(Data)
   14.842506526863986
```

Exponents and Logarithms : log, log10, exp, e, pow(x,y), sqrt, by import math module these functions are worked.

```
>>> from math import log, log10, exp, e, pow, sqrt
>>> log(3.0*e**3.4)
        4.4986122886681095
>>> sqrt(625)
        25
>>> 5**3
        125
>>> pow(6,2)
        36.0
```

Trigonometry: By import the sin function from the **math** module which is part of the Python Standard Library.sine, cosine, and tangent

```
>>> from math import sin, cos, tan, pi
>>> pi
        3.141592653589793
>>> sin(90)
        0.8939966636005579
>>> cos(0)
        1.0
>>> tan(60)
        0.320040389379563
```

3. i) Write a program to calculate compound interest when principal, rate and number of periods are given.
Program:

```
P = float(input("Enter the Principal amount: "))
R = float(input("Enter the rate of interest: "))
T = float(input("Enter the time in years: "))
CI = P * ((1 + R/100) ** T) – P
```

```
print("The compound interest is:", CI)
```

Output

```
Enter the principle amount: 5000
Enter the interest rate: 4
Enter the time period in years: 3
The  compound interest is: 105.1
```

3.ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points

Program:

```
x1 = float(input('Enter x1: '))
y1 = float(input('Enter y1: '))
x2 = float(input('Enter x2: '))
y2 = float(input('Enter y2: '))

d = ( (x2-x1)**2 + (y2-y1)**2 ) ** 0.5   # Calculating distance
print('Distance = %f' %(d))
```

Output :

```
Enter x1:  2
Enter y1:   4
Enter x2:   8
Enter y2:  10
Distance = 8.485281Enter x2
Enter x2
```

4. Read name, address, email and phone number of a person through keyboard and print the details.

Program:

```
Nam=input("Enter Name: ")
Adress=input("Enter Adrees:  HNo, Col, City, Dist, State, Pin ")
Mail=str(input("Enter email id   "))
Ph_no=int(input("Enter phone number :  "))
print( "\n Bio Data :")
print("_____")
print( "Name is      : " ,Nam, "\t\tMail Id is : " ,Mail)
print("Adress is : ", Adress)
print( "Pho Num is : " ,Ph_no)
```

Output :

Enter Name : Ravi kumar
Enter Adrees: HNo, Col, City, Dist, State, Pin 1-21,Ngos, hyderabad, Ranagareddy, 501506
Enter email id Ravi@gmail.com
Enter phone number : 8889654321

Bio Data :

_____
Name is : Ravi kumar Mail Id is : Ravi@gmail.com
Adress is : 1-21,Ngos, hyderabad, Ranagareddy, 501506
Pho Num is : 8889654321

Week - 2:

1. Print the below triangle using for loop.

```
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

Program:

```
n = int(input("Enter number of rows: "))
for r  in range(n,0,-1):
    for  c in range(r, n+1,1):
        print(r, end=" ")
    print(" ")
```

output :

```
Enter number of rows:  5
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)

Program:

```
ch=input("Enter charater:  ")
if(ch.isalpha()):
    if (ch.islower()):
        print("Lower case character ")
    else:
            print("Upper case character ")
elif(ch.isdigit()):
    print("It is Digit")
else:
    print("It is specil character")
```

Output:

```
Enter charater:  6
It is Digit

Enter charater:  r
Lower case character

Enter charater:  T
Upper case character

Enter charater:  @
```

It is specil character

3. Python Program to Print the Fibonacci sequence using while loop

Program:

```
n = int(input(" Enter range for Fibonacci:  "))
a = 0
b = 1
c=0
while(c <= n):
    print(c, end=", ")
    a=b
    b=c
    c = a + b
```

Output

```
Enter Range for Fibonacci:  20
0, 1, 1, 2, 3, 5, 8, 13,
```

4. Python program to print all prime numbers in a given interval (use break)

Program:

```
l=int(input(" enter lower limit: "))
h=int(input(" enter uper limit: "))
print(" prime numbers given range: ")
for n in range(l, h):
   for i in range(2, int(n/2)+1):
      if (n % i) == 0:
         break
   else:
      print(n, end=",  ")
```

Output :

```
Enter lower limit: 10
Enter Uper limit: 40
Prime numbers Given Range:
11,  13,  17,  19,  23,  29,  31,  37,
```

Week - 3:

1. i) Write a program to convert a list and tuple into arrays.

Program:

```
import numpy as np
List1 = [3, 4, 5, 6]
print(type(list1))
print(list1)
print()
 # conversion
Array1 = np.asarray(list1)
print(type(array1))
print(array1)
```

Output:

```
<class 'list'>
[3, 4, 5, 6]
<class 'numpy.ndarray'>
[3 4 5 6]
```

1.ii) Write a program to find common values between two arrays.

Program:

```
A = [ 2, 3, 4, 5,7,8]
B = [1, 2,5, 6, 8, 9]
print("Common Elements of two Arrays: ")
for v  in A :
    If  v  in B:
        print(v)
```

Output

```
Common Elements of two Arrays:
2
5
8
```

2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.

Program:

```
a=int(input("Enter a value "))
b=int(input("Enter b value "))
if b==0:
```

```
    print(a)

while( b>0):
    r=a%b
    a=b
    b=r
print("The gcd of  given numbers: ", a)
```

Output

```
Enter a value 8
Enter b value 12
The gcd of  given numbers:   4
```

3. Write a function called palindrome that takes a string argument and returnsTrue if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.
Program:

```
def   isPalindrome(str):
    n=int(len(str))
    for i in range(0, n//2):
        if str[i] != str[n-1-i]:
            return false
    return true
s = input(" enter string to check polindrom or not:   ")
ans = ispalindrome(s)
if (ans):
    print(" Given String Polindrome ")
else:
    print("Not Polindrome")
```

Output:

```
Enter string to check polindrom or not
 katak
Given String Polindrome

Enter string to check polindrom or not
Raju
Not Polindrome
```

Week - 4:

1. Write a function called is_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.

Program:

```
def is_sorted(L):
    for i in range(len(L)-1):
        if(L[i]>L[i+1]):
            return False
        else:
            s=True
    return s

#main program
A=[ ]
print("Enter  5 Values to check sorted or Not ")
for k in range(5):
    v=int(input())
    A.append(v)
res=is_sorted(A)
if(res):
    print(res, ", List  is sorted")
else:
    print(res, ", List  not is sorted")
```

Output:

```
Enter  5 Values to check sorted or Not
2, 5, 7, 8, 9
True , List  is sorted
Enter  5 Values to check sorted or Not
1, 2, 5, 8, 2
False , List  not is sorted
```

2. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.

Program:

```
L = [1, 2, 3, 4, 2, 7, 8, 8, 3, 5,6]
print("The List is ", L)
print("Duplicate elements are  in List:  ");
for i in range(0, len(L)):
    for j in range(i+1, len(L)):
        if(L[i] == L[j]):
            print(L[j])
```

Output:

```
The List is  [1, 2, 3, 4, 2, 7, 8, 8, 3, 5, 6]
Duplicate elements are  in List:
2
3
8
```

2.i). Write a function called remove_duplicates that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.

Program:

```
def Remove_Duplicate(A):
    B=[ ]
    for i  in A:
        if i  not in B :
            B.append(i)
    return B

L = [5, 4, 2,17, 3, 5, 2, 7, 4]
print("Befor Remove :", L)
print("List after Remove Dulicate: ", Remove_Duplicate(L))
```

Output :

```
Befor Remove : [5, 4, 2, 17, 3, 5, 2, 7, 4]
List after Remove Dulicate:  [5, 4, 2, 17, 3, 7]
```

2.ii). The wordlist I provided, words.txt, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.

Program:

```
line=input("Enter Sentence:  ")
words=line.split()
c=0
n= len(words)
for w in words:
    if(len(w) ==1):
        c=c+1
if (c==0):
    print("There is no single letter words: ")
ch=input(" Enter character to insert:  ")
p=int(input(" Enter position where to insert: "))
g=" "

for i in range(0, n):
    if(i==p-1):
      g+=ch+" "+words[i]
    else:
        g+=" "+words[i]+" "
```

```
print("After Inserting Letter:  ", g)
```

Output:

Enter Sentence: He is good boy
There is no single letter words:
Enter character to insert: a
Enter position where to insert: 3
After Inserting Letter: He is **a** good boy

Enter Sentence: am working in college
There is no single letter words:
Enter character to insert: I
Enter position where to insert: 1
After Inserting Letter: I am working in college

2.iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.

Program:

```
Dict1={  }
for Num in range(101,104):
    name = input("Enter Name: ")
    Dict1[Num] = name

print ("Original dictionary is : ")
print(Dict1)

print()
Dict2 = { }
for   key,   value  in  Dict1.items():
     if( value  in Dict2):
         Dict2[value].append(key)
     else:
        Dict2[value]=[key]

print ("Dictionary after swapping is :  ")
print("keys:   values")
for i in Dict2:
    print(i, " :", Dict2[i])
```

Output:

```
Enter Name:   Raju
Enter Name:   Vinay
Enter Name:   Srinu
```

Original dictionary is :
{101: 'Raju', 102: 'Vinay', 103: 'Srinu'}

Dictionary after swapping is :
keys: values
Raju : [101]
Vinay : [102]
Srinu : [103]

3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
Program:

```
string = input("Enter a string:  ")
n=len(string)
comma = ","
res = ""
for i in range(0, n,1):
    if(i<n-1):
       res += string[i]+comma
    else:
        res+=string[i]
print(res)
```

Output:

```
Enter a string: College
C,o,l,l,e,g,e
```

ii) Remove the given word in all the places in a string?
Program:

```
Text = input("Enter a Sentence: ")
Words = Text.split()
data = input(Enter a word to delete: ')
a=False
for  w  in Words:
    if (w == data):
        Words. remove(w)
        a=True
if(a):
    print("After Remeve word:  ", Words)
else:
     print(data, "  word is not in sentence")
```

Output:
Enter a Sentence: i am not studying
Enter a word to delete: not
After Remeve word: ['i', 'am', 'studying']

Enter a Sentence: I am writing python exam
Enter a word to delete: studying studying word is not in sentence

iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?

Program

```
Line = input("Enter sentence to Chang of each word first letter capital: ")
words = Line.split()
res = ""
for w in words:
    res = ""
    if ord (w[0]) >=97 and ord(w[0]) <= 122:
          res=res+(chr( ord(w[0]) - 32))
    else:
        res=res+w[0]
    for i in range(1, len(w)):
       if ord (w[i]) >= 65 and ord(w[i]) <= 90:
          res=res+(chr( ord(w[i]) + 32))
       else:
        res=res+w[i]
    print(res, end=" ")
```

Output

Enter sentence to Chang of each word 1st letter capital:
i am doing work in college
I Am Doing Work In College

4. Writes a recursive function that generates all binary strings of n-bit length

```
def Binary_string(n):
   legnth=2**n
   for i in range(legnth):
        binary_str = format(i, '0' + str(n) + 'b')
        print(binary_str)

n = int(input("Enter binary Number String length: "))
Binary_string(n)
```

Output:

```
000
001
010
011
100
```

101
110
111

Week - 5:

1. i) Write a python program that defines a matrix and prints

Program:

```
m = int(input("Enter the number of rows:  "))
n = int(input("Enter the number of columns: "))

# Initialize matrix
Matrix = [ ]
for r in range(m):
    a=[]
    print("Enter elements of ", r+1," row")
    for c in range(n):
        a.append(int(input()))
    Matrix.append(a)

print("The entered matrix is: "
for r in range(m):
    for c in range(n):
        print(Matrix[r][c], end="   ")
    print()
```

Output:

```
Enter the number of rows:  2
Enter the number of columns: 3
Enter elements of  1  row
5
1
3
Enter elements of  2  row
7
3
8
The entered matrix is :
5    1    3
7    3    8
```

ii) Write a python program to perform addition of two square matrices

```
A = [[1, 3], [7, 2]]
B = [[4, 6], [3, 1]]
m=len(A)
n=len(A)
Mat = [[0, 0], [0, 0]]
for  r   in   A:
```

```
        print(r)
    print()
    for  r  in   B:
        print(r)
    print("Addition above matrixes: "   )
    for   r  in   range(m):
         for  c in   range(n):
             Mat[r][c] = A[r][c]+ B[r][c]
     for  r  in   Mat:
        print(r)
```

Output:

```
[1, 3]
[7, 2]

[4, 6]
[3, 1]
Addition above matrixes:
[5, 9]
[10, 3]
```

iii) Write a python program to perform multiplication of two square matrices

Program:

```
    Mat = [ [0, 0], [0, 0] ]
    A = [ [1,   3], [5,   2] ]
    B = [ [3,   0], [2,  4] ]

    print("A: ")    #display of Matrix A
    for r in A:
       print("   ", r)

    print("B: ")  #display of Matrix B
    for r in B:
       print("   ", r)

    for r in range(len(A)):
         for c in range(len(B[0])):
              for k in range(len(B)):
                  Mat[r][c] += A[r][k] * B[k][c]

    print("Multplication: ")   #display of Multiplication of A and B
    for r in Mat:
       print("   ",r)
```

Output:

```
A:    [1,  3]
      [5,  2]

B:    [3,  0]
      [2,  4]
```

```
Multplication:
      [9,   12]
      [19,  8]
```

2. Use the structure of exception handling all general purpose exceptions.
Program:

```
try:
    a=9/0
except Exception as e:
    print("Error is :  ", e)
try:
   a=b+9
except Exception as e:
    print("Error is :  ", e)
try:
   a=[10,20,30]
   s=a[4]
except Exception as e:
    print("Error is :  ", e)
try:
   a=45+"Raju"
except Exception as e:
    print("Error is :  ",e)
```

Output:

```
Error is :   division by zero
Error is :   name 'b' is not defined
Error is :   list index out of range
Error is :   unsupported operand type(s) for +: 'int' and 'str'
```

Week-6:

1. a. Write a function called draw_rectangle that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.

Program:

```
from tkinter import *

def draw_rectangle(w, x1, y1, x2, y2):
    w.pack()
    w.create_rectangle(x1, y1, x2, y2)

#main program
print("Enter cordinates : ")
a=int(input("Enter x1: "))
b=int(input("Enter y1: "))
c=int(input("Enter x2: "))
d=int(input("Enter y2: "))
master=Tk()
master.title("Rectangle")
Cv=Canvas(master,  width=200,  height=100) #  canvas means window size
draw_rectangle(Cv,a, b, c, d)  #calling function with rectangle cordinates and
convas

if( (a-b)==(c-d) ):
   print("Above is Square ")
else:
   print("Above is Rectangle ")
```

Output:

Enter cordinates :
Enter x1: 20
Enter y1: 20
Enter x2: 50
Enter y2: 30

Above is Rectangle

b. Add an attribute named color to your Rectangle objects and modify draw_rectangle so that it uses the color attribute as the fill color.

```
from tkinter import *
```

```
def draw_rectangle(win, x1, y1, x2, y2, col):
    win.pack()
    win.create_rectangle(x1, y1, x2, y2, fill='red')

#main program
print("Enter cordinates : ")
a=int(input("Enter x1: "))
b=int(input("Enter y1: "))
c=int(input("Enter x2: "))
d=int(input("Enter y2: "))
master=Tk()
master.title("Rectangle")
Cv=Canvas(master,  width=200,  height=100) #  canvas means window size
draw_rectangle(Cv,a, b, c, d)  #calling function with rectangle cordinates and
convas

if( (a-b)==(c-d) ):
    print("Above is Square ")
else:
    print("Above is Rectangle ")
```

Output:

```
Enter cordinates :
Enter x1: 20
Enter y1: 20
Enter x2: 50
Enter y2: 30
```

```
Above is Rectangle
```

c. Write a function called draw_point that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.

Program:

```
from tkinter import *
from tkinter import ttk
def draw_point(event):
  x1=event.x
  y1=event.y
  x2=event.x
  y2=event.y
  canvas.create_oval(x1,y1,x2,y2,fill="red" , width="10")
```

```
# Create a canvas widget
win=Tk()
win.title("point")
canvas=Canvas(win, width=700, height=350, background="white")
canvas.grid(row=0, column=0)
canvas.bind('<Button-1>', draw_point)

win.mainloop()
```

Output:

d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called draw_circle that draws circles on the canvas.

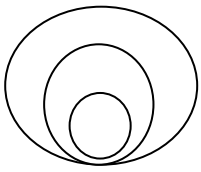
Program:

```
import turtle
def draw_circle(n):
    t = turtle.Turtle()
    print("Enter ",n, " circle radius : ")
    for i in range(1,n+1,1):
        r=int(input())
        t.circle(r)

a=int(input("Enter how many circles to draw:  "))
draw_circle(a)
```

Output:

```
Enter how many circles to draw:  3
Enter  3  circle radius :
10
30
20
```



2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.

```
class A:
   def func(self):
      print("I am a class A")

class B(A):
   def func(self):
      print("I am a class B")
```

```
class C(A):
    def func(self):
        print("I am a class C")
class D(B,C):
    def func(self):
        print("I am a class C")
d = D()
print(d.func())
b = B()
print(b.func())
c = C()
print(c.func())
a = A()
print(a.func())
```

Output:

```
I am a class C
None
I am a class B
None
I am a class C
None
I am a class A
None
```

3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Program:

```
import re
mail  =  input("Enter main Id to valid or not: ")
phn  =  input('Enter Mobile number :')

pat   = r'\b[A-Za-z0-9._]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b'  #mail id pattern
if(re.fullmatch(pat, mail)):
      print(mail  , "Valid Email")
else:
      print(mail  , "Invalid Email")

res   =  re.fullmatch('[6-9][0-9]{9}',phn)   # [6-9] means first digit any of 6,7,8,9 , next {9} digits 0 -9

if (res!=None):    # checking whether it is none or not
    print(phn, ": Valid phone Number")
else:
    print(phn , "  : Invalid phone number")
```

Output:

```
Enter main Id to valid or not: Raju_kumar@gmail.com
Enter Mobile number :6983340921
```

Raju_kumar@gmail.com: Valid Email
6983340921 : Valid phone Number

Week- 7

1. Write a Python code to merge two given file contents into a third file.

Program:

```
data1 = data2 = "";
fp=open('file1.txt',  'r') # Reading data from file1
data1 = fp.read()
print('file 1 data is :', data1)

fp=open('file2.txt',  'r')   # Reading data from file2
data2 = fp.read()
print('file 2 data is: ', data2)

data1+= "\n"      # Merging 2 filesTo add the data of file2 from next line
data1 += data2

with open('file3.txt', 'w+') as fp:  #inserting merged data into file3
   fp.write(data1)

with open('file3.txt', 'r') as fp:
     data3=fp.read()
print(' After Merge File 3 data is:  ' , data3)
```

Output:

file 1 data is : Hi this is first file python is a better language.
file 2 data is: as student has to learn python properly to get software job

After Merge File 3 data is: Hi this is first file python is a better language as student has to learn python properly to get software job

2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.

Program:

```
fp=open('file1.txt',  'r') # Reading data from file1
data1 = fp.read()
print('file 1 data is :', data1)
w=input(" Enter word to found: ")
#with open('file1.txt', 'r') as f:

words = data1.split()
if w in words:
    print(w, " word is found in file ")
else:
    print("word is Not found in file ")
```

Output:

file 1 data is : Hi this is first file python is a better language .
Enter word to found: python
Python word is found in file
file 1 data is : Hi this is first file python is a better language
Enter word to found: java
Java word is Not found in file

3. Write a Python code to Read text from a text file, find the word with most number of occurrences
Program:

```
fp=open('file1.txt',  'r')    # Reading data from file1
data1 = fp.read()
print('file 1 data is :', data1)
w=input(" Enter word to found: ")
#with open('file1.txt', 'r') as f:
k=0
words = data1.split()
for i  in words:
    if( i==w):
        k=k+1
print("given word occurred times: ",  k)
```

Output:
file 1 data is : Python Hi this is Python programming. Python is interpreted language.
Enter word to found: Python
given word occured times: **3**

4. Write a function that reads a file file1 and displays the number of words, number of vowels, blank spaces, lower
case letters and uppercase letters.
Program:

```
def counting():
    fp = open('file1.txt', "r")
    data=fp.read()
    w=data.split()
    print("The text in File is :",data )
    Vowels = L_case = U_case = Digits=Spaces=Special=0

    Vowels_List  = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']
    for  ch  in  data:
        if   ch  in  Vowels_List:
            Vowels += 1
```

```
            if( ch.islower() ):
                L_case+=1
            elif( ch.isupper() ):
                U_case+=1
            elif(ch.isdigit()):
               Digits+=1
            elif(ch==" "):
               Spaces+=1
            else:
               Special+=1

        print("Total words in file= " ,len(w))
        print("Number of vowels in file = ", Vowels)
        print("Lower case letters file= " , L_case)
        print("Upper case letters file= " , U_case)
        print("Digits   file= " , Digits)
        print("Blank spaces file= " , Spaces)
        print("Special letters=  " , Special)

    print("Program to count    vowels, lower, upper case, words in file: " )
    counting()
```

Output:

Program to count vowels, lower, upper case, words in file:

The text in File is : I am Learning Python, C&Os Language in depthly in college by paid fee =55990

Total words in file= 16
Number of vowels in file = 21
Lower case letters in file= 51
Upper case letters in file= 6
Digits in file= 5
Blank spaces file= 16
Special letters= 3

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.

Program:

```
from scipy.special import  *
 import numpy as np

print("Scipy Moudle Functions : ")
print("cube roor of 125  :  ",  cbrt(125) )
print("Exponent of 10    :  ",  exp10(3) )
print("Experl of 100       :  ", exprel(0))
print("Permetutions of 5, 2 :  ",  perm(5, 2))
print("combinations of 5, 2 :  ",  comb(5,2))
print("gamma of 65       :  ",  gamma(65) )
print("lambertw of 75    :  ",  lambertw(75) )
a = [11, 22, 33, 54, 65]
print("  logsumexp  of  [11, 22, 33, 54, 65]:   ",  logsumexp(a) )
```

Output

```
Scipy Moudle Functions :
cube roor of 125    :   5.0
Exponent of 10      :   1000.0
Experl of  100       :    2.6881171418161357e+41
Permetutions of 5, 2 :   20.0
combinations of 5, 2 :   10.0
gamma of 65           :   1.2688693218588415e+89
lambertw of 75         :   (3.1652546143148985+0j)
logsumexp  of  [11, 22, 33, 54, 65]:    65.00001670156134
```

2. a) Install NumPy package with pip and explore it.

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python**.**

Step 1: Launch the Terminal/Command Prompt Start by launching the terminal on Linux and macOS or the Command Prompt (cmd) on Windows.

Step 2: Verify Python Installation:
Run command **:** **python --version**

Step 3: Update Pip Type pip -V or pip3 -V in the terminal.
Run command to update pip: **python -m pip install --upgrade pip**

Step 4: Install NumPy in Python:
Run command in the terminal: **pip install numpy**

Step 5: Verify NumPy Installation After installation
Run command: **pip show numpy**

Program:

```
import numpy as np
array1 = np.array([3, 5, 8, 12, 15, 25])
print("Original array:\n", array1)
arr_sum=np.sum([3, 5, 8, 12, 15, 25])
min_value = np.min(array1)
print("Minimum Value:", min_value)

max_value = np.max(array1)
print("Maximum Value:", max_value)
print("\nSum of arrays: ", arr_sum)
array2 = np.reshape(array1, (2, 3))
array3 = np.transpose(array2)
print("\nReshaped array:\n", array2)
print("\nTransposed array:\n", array3)
```

Output:

```
Original array:
 [ 3  5  8 12 15 25]
Minimum Value: 3
Maximum Value: 25

Sum of arrays:  68
Reshaped array:
 [[ 3  5  8]
 [12 15 25]]

Transposed array:
 [[ 3 12]
 [ 5 15]
 [ 8 25]]
```

3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
Program:

```
def AND_Gate(a, b):
    if (a == 1 and b == 1):
        return True
    else:
        return False
def OR_Gate(a, b):
    if (a == 1 or b == 1):
        return True
    else:
        return False
def NOT_Gate(a, b):
    if (a == 1 ):
```

```
            return True
        else:
            return False
    def XOR_Gate(a, b):
        if (a !=b):
            return True
        else:
            return False
    print(" AND_gate\t  OR_gate  \t   NOT_gate \t     XOR_gate")

    for i in range(2):
        for j in range(2):
            print(i," AND ",j , ": " , AND_Gate(i, j),   end="   ;  ")
            print(i," OR   ",j , ": ", OR_Gate(i, j) ,    end="   ;   ")
            print(i," NOT ",j , ": " ,NOT_Gate(i, j),   end="   ;   ")
            print(i," XOR ",j , ": ", XOR_Gate(i, j) ")
            print('\n')
```

Output:

AND_gate	OR_gate	NOT_gate	XOR_gate
0 AND 0 : False ;	0 OR 0 : False ;	0 NOT 0 : False ;	0 XOR 0 : False
0 AND 1 : False ;	0 OR 1 : True ;	0 NOT 1 : False ;	0 XOR 1 : True
1 AND 0 : False ;	1 OR 0 : True ;	1 NOT 0 : True ;	1 XOR 0 : True
1 AND 1 : True ;	1 OR 1 : True ;	1 NOT 1 : True ;	1 XOR 1 : False

4. Write a program to implement Half Adder, Full Adder, and Parallel Adder

Program:

```
    def half_adder():
        print("Half Adder : ")
        print("A   B   Sum   Carry ")
        for  A in  range(2):
            for B in range(2):
                Sum=A^B
                Carry=A&B
                print ( A,"  ",B, "   ", Sum, "      ", Carry)

    def full_adder():
        print("Full Adder : ")
        print("A   B   C_IN  Sum   Carry ")
        for  A in  range(2):
            for B in range(2):
                for C in range(2):
                    Sum =C ^ (A ^ B)
                    Carry =(A&B) | (B&C) | (C&A)
                    print (A,"  ",B, "    ",C, "    ",Sum, "         ",Carry)

    half_adder()
    full_adder()
```

Output:

Half Adder :

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full Adder:

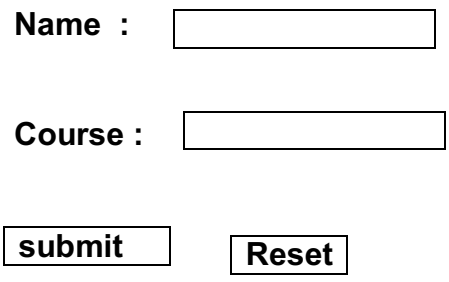
A	B	C_IN	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

Program:

```
from tkinter import *
win = Tk()  # create root window
win.title("Welcome to MCA")
win.geometry('350x200')
lbl = Label(win, text = "Name  : ")   # adding a label to the root window
lbl.grid()
lbl = Label(win, text = "Course : ")
lbl.grid()
txt = Entry(win, width=15)     # adding Entry Field
txt.grid(column =1, row =0)
txt = Entry(win, width=15)
txt.grid(column =1, row =1)
 sub = Button(win, text = " Submit" , fg = "blue") # button widget with red color text
inside
res = Button(win, text = "  Reset " , fg = "blue")
sub.grid(column=2, row=2)  # Set Button Grid
res.grid(column=3, row=2)
```

Output:



www.srichaitanyaengg.com
E-mail : director8a.sctc@gmail.com

SRI CHAITANYA TECHNICAL CAMPUS

COLLEGE OF ENGINEERING & TECHNOLOGY
COLLEGE OF BUSINESS MANAGEMENT

(Approved by AICTE, NEW DELHI & Affiliated to JNTU, Hyderabad)

Sheriguda (V), Ibrahimpatnam (M), R.R. Dist. - 501 510 - A.P.

Ph : 08414 - 223222, 223223 Fax : 08414 - 222678