



PNS SCHOOL OF ENGINEERING & TECHNOLOGY
Nishamani Vihar, Marshaghai, Kendrapara

LAB MANUAL FOR
PYTHON PROGRAMMING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
5TH SEMESTER

PREPARED BY
MR. BISWARANJAN SWAIN
LECTURER IN COMPUTER SCIENCE & ENGINEERING

1. Write a program to demonstrate different number datatypes in python.

Source code:

```
i=7
c=24+8j
f=701
s='HELLO EVERYONE!!\nThis is john\'s python programming..'
# NOTE: boolean has truth values that are case sensitive Ex: True (T is caps!)
b= True

print("the value of c is:",i,'\nits type is:',type(i))
print("the value of c is:",f,'\nits type is:',type(f))
print("the value of c is:",c,'\nits type is:',type(c))
print("the value of c is:",s,'\nits type is:',type(s))
rint("the value of c is:",b,'\nits type is:',type(b))

print('NOTE: boolean has truth values that are case sensitive Ex: True (T is caps!)
```

Output:

```
====
== RESTART: C:/Users/dell/Desktop/python programs/demonstrate_datatypes.py ==
the value of c is: 7
its type is: <class 'int'>
the value of c is: 7.01
its type is: <class 'float'>
the value of c is: (24+8j)
its type is: <class 'complex'>
the value of c is: HELLO EVERYONE!!
This is john's python programming..
its type is: <class 'str'>
the value of c is: True
its type is: <class 'bool'>
NOTE: boolean has truth values that are case sensitive Ex: True (T is caps!)
====
```

2. Write a program to perform different arithmetic operations on numbers in python.

Source code:

```
a=10; b=3  
  
print("addition of a:",a,"&b:",b,"is:",a+b)  
print("substraction of a:",a,"&b:",b,"is:",a-b)  
print("multiplication of a:",a,"&b:",b,"is:",a*b)  
print("division of a:",a,"&b:",b,"is:",a/b)  
print("floor divison of a:",a,"&b:",b,"is:",a//b)  
print("moduli of a:",a,"&b:",b,"is:",a%b)  
print("exponent of a:",a,"&b:",b,"is:",a**b)
```

Output:

```
...  
== RESTART: C:/Users/dell/Desktop/python programs/arithmetic_operations.py ==  
addition of a: 10 &b: 3 is: 13  
substraction of a: 10 &b: 3 is: 7  
multiplication of a: 10 &b: 3 is: 30  
division of a: 10 &b: 3 is: 3.3333333333333335  
floor divison of a: 10 &b: 3 is: 3  
moduli of a: 10 &b: 3 is: 1  
exponent of a: 10 &b: 3 is: 1000
```

3. Write a program to create, concatenate and print a string and accessing substring from a given string.

Source code:

```
pi=3.14
s= "Venkata"
v= "Subhramanyam"

print("the value of s is:",s)

print("the value of v is:",v)

string_add = s+v

print("after concatenating s and v the string is:",s+v)

text = 'The value of pi is ' + str(pi)

print("NOTE: variables after '+' operator must be converted to string before using them as strings\
otherwise value will be considered as its class type")

print(text)
```

Output:

```
==== RESTART: C:/Users/dell/Desktop/python programs/string_operations.py ====
the value of s is: Venkata
the value of v is: Subhramanyam
after concatenating s and v the string is: VenkataSubhramanyam
NOTE: variables after '+' operator must be converted to string before using them as strings
other wise value will be considered as its class type
The value of pi is 3.14
```

4. Write a python script to print the current date in following format “Sun
May29 02:26:23 IST 2017”

Source code:

```
import time
import datetime
x =datetime.datetime.now()
print(x.strftime("%c"))
```

Output:

```
===== RESTART: C:/Users/dell/Desktop/python programs/printing_date.py =====
Mon Jul 29 20:59:01 2019
|
```

5. Write a python program to create, append and remove lists in python.

Source code:

```
# creating list with college names..

colleges = ["SIIET", "GNIT", "AVN"]

print(colleges)

# appending new college in collges list

colleges.append("MVSR")

#checking if its added or not

print(colleges)

#adding a new college at a positon

colleges.insert(1,"BHARAT") print(colleges)

#remove a name from colleges

colleges.remove("BHARAT")

print(colleges)

#remove a name with an index value

del colleges[1]

# NOTE: index starts from 0 so 2nd value in list will be removed

print(colleges)
```

Output:

```
==== RESTART: C:/Users/dell/Desktop/python programs/lists.py =====
['SIIET', 'GNIT', 'AVN']
['SIIET', 'GNIT', 'AVN', 'MVSR']
['SIIET', 'BHARAT', 'GNIT', 'AVN', 'MVSR']
['SIIET', 'GNIT', 'AVN', 'MVSR']
['SIIET', 'AVN', 'MVSR']
```

6. Write a program to demonstrate working with tuples in python

Source code:

```
# creating tuples with college names..

colleges = ("SIJET","BHARAT","GNIT", "AVN")

print("the lists in colleges tuple is",colleges)

print("we can\t add or remove new elements in a tuple")

print("length of the tuple colleges is:",len(colleges))

# checking whether 'SIJET' is present in the tuple or not

if "SIJET" in colleges:

    print("Yes, 'SIJET' is in the colleges tuple")
```

Output:

```
===== RESTART: C:/Users/dell/Desktop/python programs/tuples.py =====
the lists in colleges tuple is ('SIJET', 'BHARAT', 'GNIT', 'AVN')
we can't add or remove new elements in a tuple
length of the tuple colleges is: 4
Yes, 'SIJET' is in the colleges tuple
```

7. Write a program to demonstrate working with dictionaries in python

Source code:

```
# creating a dictionary
for SIJET college = {
    "name": "siiet",
    "code": "INDI",
    "id": "x3"
}
print(college)
#adding items to dictionary
college["location"] = "IBP"
print(college)
#changing values of a key
college["location"] = "sheriguda"
print(college)
# to remove items
use pop()
college.pop("code")
print(college)
#know the length using len()
print("length of college is:",len(college))
#to copy the same dictionary use copy()
mycollege= college.copy()
print(mycollege)
```

Output:

```
===== RESTART: C:/Users/dell/Desktop/python programs/dictionaries.py =====
{'name': 'siiet', 'code': 'INDI', 'id': 'x3'}
{'name': 'siiet', 'code': 'INDI', 'id': 'x3', 'location': 'IBP'}
{'name': 'siiet', 'code': 'INDI', 'id': 'x3', 'location': 'sheriguda'}
{'name': 'siiet', 'id': 'x3', 'location': 'sheriguda'}
length of college is: 3
{'name': 'siiet', 'id': 'x3', 'location': 'sheriguda'}
```


8. Write a python program to find largest of three numbers

Source code:

user-defined function to know which number is

```
larger def bigOf3(a,b,c):
    if(a>b):
        if(a>c):
            print("a is greater than b and c")
        else:
            print("c is greater than a and b")
    elif(b>c):
        print("b is greater than a and c")
    else:
        print("c is greater than a and b")
txt= input("enter a,b,c values:")
a,b,c= txt.split()
bigOf3(int(a),int(b),int(c)) #calling the function
```

Output:

```
.....
===== RESTART: C:/Users/dell/Desktop/python programs/largest.py =====
enter a,b,c values:2 4 5
c is greater than a and b
>>> |
```

9. Write a python program to convert temperature to and from Celsius to fahrenheit.

Source code:

```
while(1):
    print("1.CELSIUS TO FAHRENHEIT\n2.FAHRENHEIT TO CELSIUS\n3.EXIT\n")
    choice=input("ENTER YOUR CHOICE:")
    ch=int(choice)
    if(ch==1):
        c=int(input("ENTER TEMPERATURE IN CELSIUS:"))
        f=((9*c)/5)+32
        print("converted temperature is:",f)
    elif(ch==2):
        f=int(input("ENTER TEMPERATURE IN FAHRENHEIT:"))
        c=((f-32)/9)*5
        print("converted temperature is:",c)
    elif(ch==3):
        exit()
    else:
        print("wrong choice")
```

Output:

```
===== RESTART: C:/Users/dell/Desktop/python programs/temperature.py =====
1.CELSIUS TO FAHRENHEIT
2.FAHRENHEIT TO CELSIUS
3.EXIT

ENTER YOUR CHOICE:1
ENTER TEMPERATURE IN CELSIUS:32
converted temperature is: 89.6
1.CELSIUS TO FAHRENHEIT
2.FAHRENHEIT TO CELSIUS
3.EXIT

ENTER YOUR CHOICE:2
ENTER TEMPERATURE IN FAHRENHEIT:89
converted temperature is: 31.666666666666664
1.CELSIUS TO FAHRENHEIT
2.FAHRENHEIT TO CELSIUS
3.EXIT

ENTER YOUR CHOICE:3
```

10. Write a python program to construct the following pattern using nested for loop:

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

Source code:

```
n=int(input("ENTER A VALUE:"))
```

```
for x in range(0,n+1,1):
```

```
    print(x**x')
```

```
    if(x==n):
```

```
        for x in range(n,0,-1): print(x**x')
```

Output:

```
===== RESTART: C:/Users/dell/Desktop/python programs/pattern.py =====
ENTER A VALUE:5
*
**
***
****
*****
*****
*****
****
***
**
*
```

11. Write a python program to print prim numbers less than 20:

Source code:

```
n=int(input("enter range of prime numbers:"))

for num in range(2,n+1):                #takes each number
    count=0
    for i in range(2,num//2+1):         #checks the divisibility of each num
        if(num%i==0):
            count=count+1              #if its noot prime count increases.
    if(count==0):
        print(num)
```

Output:

```
= RESTART: C:/Users/dell/AppData/Local/Programs/Python/Python36-32/prime.py =
enter range of prime numbers:20
2
3
5
7
11
13
17
19
.
```

12. Write a python program to find factorial of a number using recursion:

Source code:

```
def recursion(n):  
  
    if(n<1):  
        print("FACTORIAL NOT POSSIBLE!!!")  
  
    elif(n>1):  
        return n*recursion(n-1)  
  
    else:  
        return 1  
  
n=int(input("enter a number:"))  
print("factorial of",n,"is:",recursion(n))
```

OUTPUT:

```
....  
RESTART: C:/Users/dell/AppData/Local/Programs/Python/Python36-32/recursion.py  
enter a number:5  
factorial of 5 is: 120  
>>> |
```

13. Write a python program to that accepts length of three sides of a triangle as inputs. The program should indicate whether or not the triangle is a right-angled triangle (use Pythagorean theorem):

Source code:

```
a=float(input("enter length of hypotenuse side:"))

b=float(input("enter length of base side:"))

c=float(input("enter length of height side:"))

def pythagorean(a,b,c):    #defining function

a=a*a; b=b*b; c=c*c

    if(a==b+c):

        print("yes!! the given inputs are triplets of a right angled triangle!!")

        print("height:",c**0.5,"\nbase:",b**0.5,"\nhypotenuse:",a**0.5)

pythagorean(a,b,c)    # calling function
```

Output:

```
....
RESTART: C:/Users/dell/AppData/Local/Programs/Python/Python36-32/triangle.py
enter length of hypotenuse side:5
enter length of base side:4
enter length of height side:3
yes!! the given inputs are triplets of a right angled triangle!!
height: 3.0
base: 4.0
hypotenuse: 5.0
>>>
```

14. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

Source code:

fibonacci.py

```
def fibonacci(n):  
    n1=0; n2=1;  
    print(n1)  
    print(n2)  
    for x in range(0,n):  
        n3=n1+n2  
        if(n3>=n):  
            break;  
        print(n3,end = ' )  
        n1=n2  
        n2=n3
```

using_fibonacci.py

Note: we will be using previous program as a library or package It is mandatory to write both the programs are separately

```
import fibonacci  
n=int(input("enter range:"))  
if(n<0):  
    print("enter correct range!!")  
else:  
    print(" ..... FIBONACCI SERIES ..... \n")  
    fibonacci.fibonacci (n)
```

Output:

```
==== RESTART: C:\Users\dell\Desktop\python programs\using_fibonacci.py ====  
enter range:30
```

```
-----FIBONACCI SERIES-----
```

```
0  
1  
1  
2  
3  
5  
8  
13  
21  
>>>
```


15. Write a python program to define a module and import a specific function in that module to another program.

Source code:

fibonacci.py

```
def fibonacci(n):  
  
    n1=0;  
    n2=1;  
    print(n1)  
  
    print(n2)  
    for x in range(0,n):  
        n3=n1+n2  
        if(n3>=n):  
            break;  
        print(n3,end = ' ')  
        n1=n2  
        n2=n3
```

using fibonacci.py

Note: we will be using previous program as a library or package It is mandatory to write both the programs are separately

```
from fibonacci import fibonacci  
n=int(input("enter range:"))  
if(n<0):  
    print("enter correct range!!")  
else:  
    print(" ..... FIBONACCI SERIES ..... \n")  
    fibonacci (n)
```

Output:

```
===== RESTART: C:\Users\dell\Desktop\python programs\using_fibonacci.py =====  
enter range:30
```

```
-----FIBONACCI SERIES-----
```

```
0  
1  
1  
2  
3  
5  
8  
13  
21  
>>>
```

16. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

Source code:

Note: create a text file as "input.txt" and write some date in it. This will be used in the program.

```
with open("input.txt") as input:
```

```
    with open("output.txt","w") as output:
```

```
        for line in input: output.write(line)
```

```
print("JOB DONE!!")
```

Output:

```
***
===== RESTART: C:/Users/dell/Desktop/python programs/copyfile.py =====
JOB DONE!!
>>> |
```

17. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.

Source code:

```
fname=input("enter file name with correct extension:")
file_opened=open(fname)
our_list=list()                                #creating an empty list
for line in file_opened:
    word=line.rstrip().split()                 #rstrip for removing unwanted spaces
for element in word:
    if element in our_list:
        continue
    else:
        our_list.append(element)
our_list.sort()
print(our_list)
```

Output:

```
== RESTART: C:/Users/dell/Desktop/python programs/printing_common_words.py ==
enter file name with correct extension:output.txt
['a', 'as', 'be', 'copied', 'data', 'file', 'input!!', 'new', 'taken', 'thank',
'this', 'to', 'was', 'will', 'you']
>>> |
```

18. Write a Python class to convert an integer to a roman numeral.

Source code:

```
class roman_solution:

    def int_to_Roman(num):

        val = [1000, 900, 500, 400, 100, 90, 50, 40, , 9, 5, 4, 1]

        syb = ["M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I" ]

        roman_num = ""

        i = 0

        if(n<1 or n>3999):

            print("ENTER A GOOD VALUE!")

        else:

            while num > 0:

                if(num-val[i]>=0):

                    roman_num+=syb[i]

                    num-=val[i]

                else:

                    i+=1

            return roman_num n=int(input("ENTER A NUMBER:"))

print("roman numeral of given number is:",roman_solution.int_to_Roman(n))
```

Output:

```
===== RESTART: C:/Users/dell/Desktop/python programs/int_to_roman.py =====
ENTER A NUMBER:2408
roman numeral of given number is: MMCDVIII
>>> |
```

19. Write a Python class to implement pow(x, n)

Source code:

```
class py_power:

    def power(x,n):

        print("power of given literals:\nx:",x,"\nn\nn:",n,"is:",x**n)

x=float(input("ENTER X(BASE) VALUE:"))

n=float(input("ENTER N(POWER) VALUE:"))

py_power.power(x,n)
```

Output:

```
.....
===== RESTART: C:/Users/dell/Desktop/python programs/power.py =====
ENTER X(BASE) VALUE:26.4765
ENTER N(POWER) VALUE:2
power of given literals:
x: 26.4765
n: 2.0
is: 701.0050522500001
>>>
```

20. Write a Python class to reverse a string word by word.

Source code:

```
fname="HELLO EVERYONE THIS IS PYTHON PROGRAMMING AND WE'RE PLAYING  
WITH LISTS"
```

```
our_list=list()                #creating an empty list
```

```
word=fname.split()            #splitting up the list
```

```
for element in word:
```

```
    our_list.append(element)
```

```
    print("tried sentence is:",our_list)
```

```
our_list.reverse()            #method to reverse the elements in the list
```

```
print("list after the reverse()",our_list)
```

Output:

```
== RESTART: C:\Users\dell\Desktop\python programs\printing_common_words.py ==  
tried sentence is: ['HELLO', 'EVERYONE', 'THIS', 'IS', 'PYTHON', 'PROGRAMMING',  
'AND', 'WE'RE', 'PLAYING', 'WITH', 'LISTS']  
list after the reverse() ['LISTS', 'WITH', 'PLAYING', 'WE'RE', 'AND', 'PROGRAMMI  
NG', 'PYTHON', 'IS', 'THIS', 'EVERYONE', 'HELLO']  
>>> |
```
