INDEX

PIRENS Institute of Business Management and Administration, Loni Bk.
Seat No: 4956 Sign:
Student Name: Vishal Shridar Vishwakarma
Subject Name: IT21L Practical
Program Title: 1) Python installation and configuration with windows and Linux.

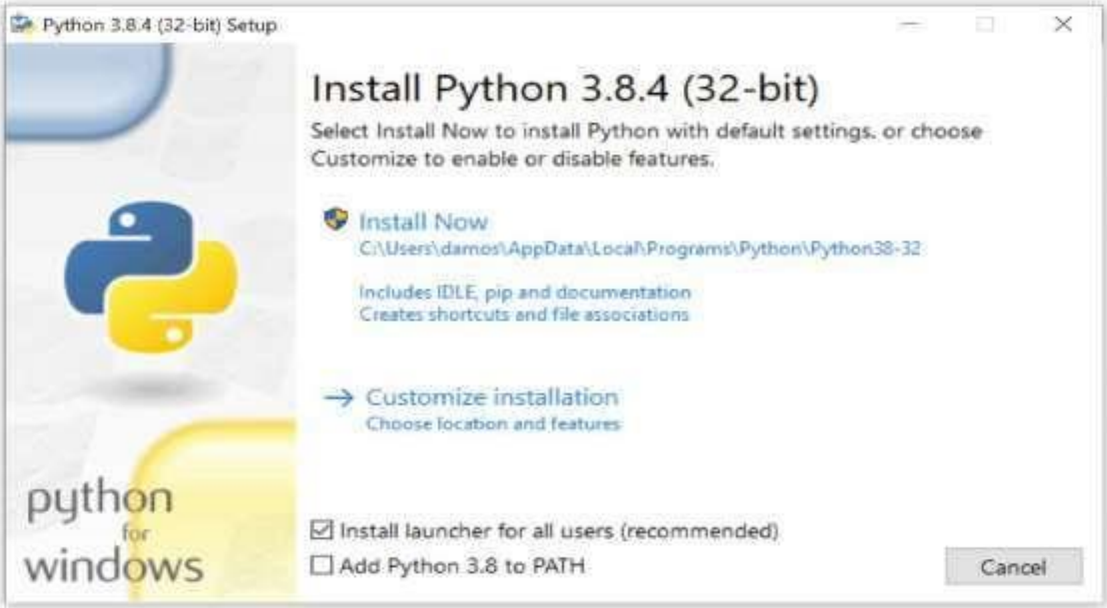
1.1. Installation steps

Four Python 3.9 installers are available for download - two each for the 32-bit and 64-bit versions of the interpreter. The *web installer* is a small initial download, and it will automatically download the required components as necessary. The *offline installer* includes the components necessary for a default installation and only requires an internet connection for optional features. See Installing Without Downloading for other ways to avoid downloading during installation.

After starting the installer, one of two options may be selected:

2: Run the Installer:

Once you've chosen and downloaded an installer, run it by double-clicking on the downloaded file. A dialog box like the one below will appear:



Python 3.8.4 (32-bit) Setup

Install Python 3.8.4 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\damos\AppData\Local\Programs\Python\Python38-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
□ Add Python 3.8 to PATH

There are four things to notice about this dialog box:

1. The default install path is in the AppData/ directory of the current Windows user.

2. The *Customize installation* button can be used to customize the installation location and which additional features get installed, including pip and IDLE.
3. The *Install launcher for all users (recommended)* checkbox is checked default. This means every user on the machine will have access to the py.exe launcher. You can uncheck this box to restrict Python to the current Windows user.
4. The *Add Python 3.8 to PATH* checkbox is unchecked by default. There are several reasons that you might not want Python on PATH, so make sure you understand the implications before you check this box.

The full installer gives you total control over the installation process.

Customize the installation to meet your needs using the options available on the dialog box. Then click *Install Now*. That's all there is to it!

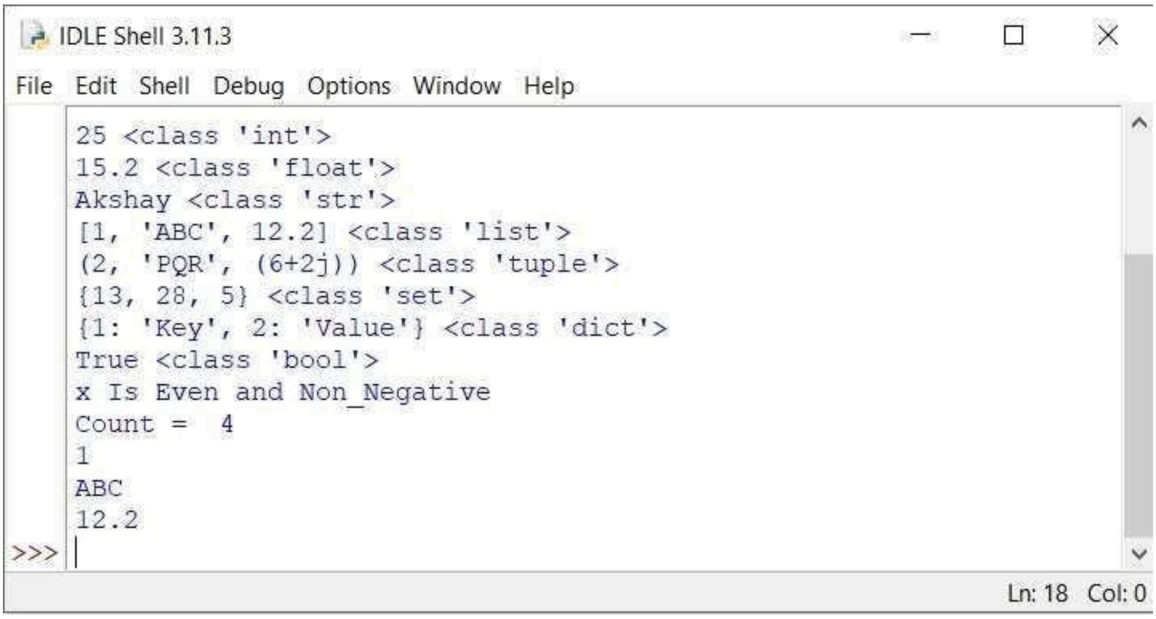
Congratulations—you now have the latest version of Python on your Windows machine!

PIRENS Institute of Business Management and Administration, Loni Bk.	
Seat No: 4 9 5 6	**Sign:**
Student Name: Vishal Shridar Vishwakarma	
Subject Name: IT21L Practical	
Program Title: 2) Programs for understanding the data types, control Flow statements, blocks and loops.	

Code :

```
          #Data types

          num = 25 print(num ,
          type(num)) no = 15.20
          print(no,type(no)) string =
          "Akshay"
          print(string,type(string))
          lst = [1,"ABC" ,12.2]
          print(lst,type(lst))
          tup=(2,"PQR",6+2j)
          print(tup,type(tup))
          set1={13,28,5}
          print(set1,type(set1))
          dict1 = {1:'Key',2:'Value'}
          print(dict1,type(dict1)) a=True
          print(a,type(a))

          #Control Statements
          x=8 if x < 0:
             print("x Is Negative")
          elif x % 2:
             print("x Is Positive and odd")
          else:
             print ("x Is Even and Non_Negative")

          #While loop
          count = 0
          while x > 0:
          x = x // 2
          count += 1
          print("Count = ", count)

          #for loop
          for i in lst:
print(i)
```

Output:



```
25 <class 'int'>
15.2 <class 'float'>
Akshay <class 'str'>
[1, 'ABC', 12.2] <class 'list'>
(2, 'PQR', (6+2j)) <class 'tuple'>
{13, 28, 5} <class 'set'>
{1: 'Key', 2: 'Value'} <class 'dict'>
True <class 'bool'>
x Is Even and Non_Negative
Count =  4
1
ABC
12.2
>>> 
```

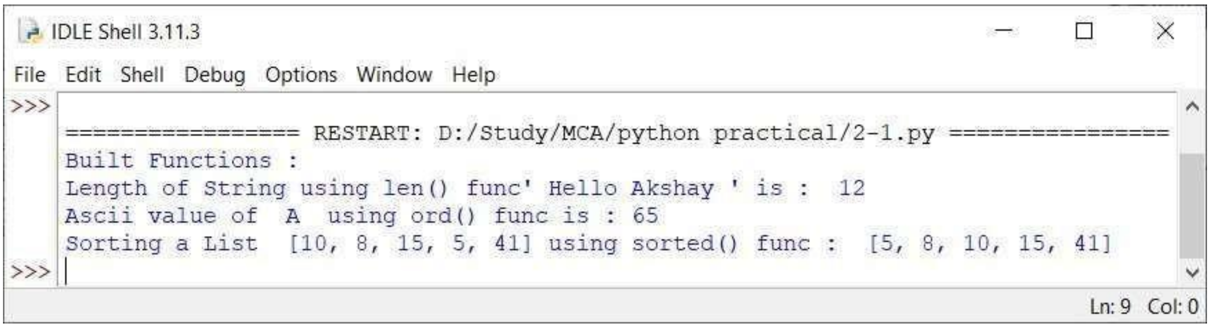
PIRENS Institute of Business Management and Administration, Loni Bk.	
Seat No: 4956	**Sign:**
Student Name: Vishal Shridar Vishwakarma	
Subject Name: IT21L Practical	
Program Title: 3) Programs for understanding functions, use of built in functions, user defined Functions	

Code:

1) **Built in Function: -**

```
print("Built Functions : ")
string = "Hello Akshay"
print("Length of String using len() func'", string, "' is : ", len(string)) c
= 'A'
print("Ascii value of ", c, " using ord() func is :", ord(c))
arr = [10, 8, 15, 5, 41]
print("Sorting a List ", arr, "using sorted() func : ", sorted(arr))
```

Output:

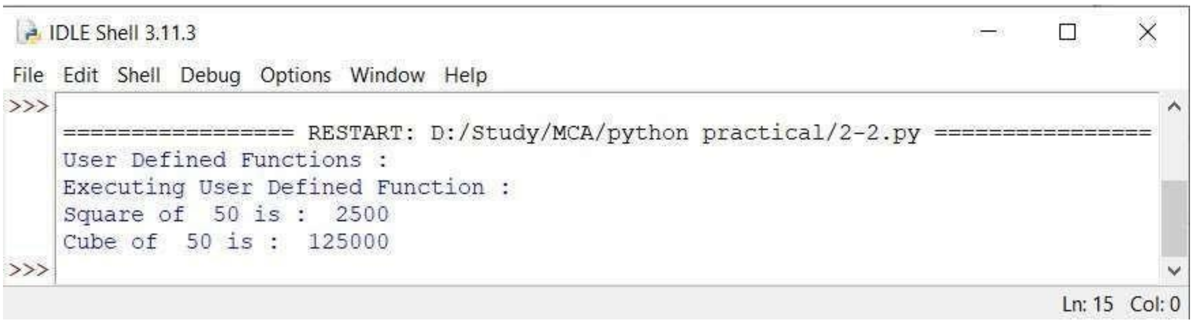


```
>>>
================ RESTART: D:/Study/MCA/python practical/2-1.py ================
Built Functions :
Length of String using len() func' Hello Akshay ' is :  12
Ascii value of  A  using ord() func is : 65
Sorting a List  [10, 8, 15, 5, 41] using sorted() func :  [5, 8, 10, 15, 41]
>>>
```

2) **User Defined Functions: -** print("User Defined Functions :

```
")def sqrandcube(number):
    print("Square of ", number, "is : ", number**2)
print("Cube of ", number, "is : ", number**3)
print("Executing User Defined Function : ")
num=50 sqrandcube(num)
```

Output:



```
>>>
    ================ RESTART: D:/Study/MCA/python practical/2-2.py ================
    User Defined Functions :
    Executing User Defined Function :
    Square of  50 is :  2500
    Cube of  50 is :  125000
>>>
```

PIRENS Institute of Business Management and Administration, Loni Bk.	
Seat No: 4956	**Sign:**
Student Name: Vishal Shridar Vishwakarma	
Subject Name: IT21L Practical	
Program Title: 4) Programs to use existing modules, packages and creating modules, packages.	

Code: module.py def

greeting(name): print("Hello "

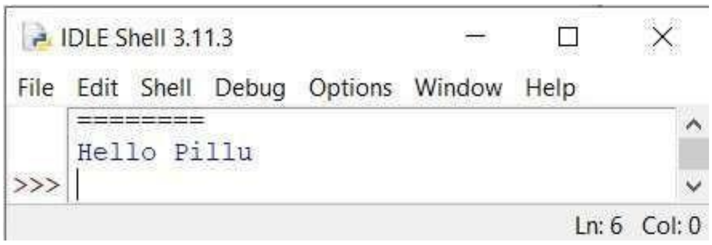
+ name)

Import the module: import

module

module.greeting("Pillu")

Output:



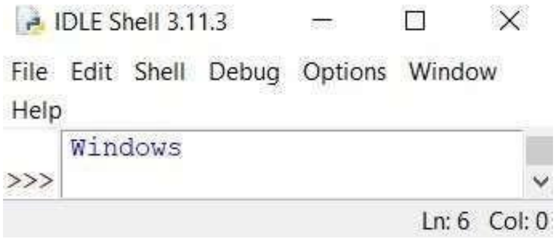
```
========
Hello Pillu
>>>
```

Built-in modules import

platform x =

platform.system() print(x)

Output:



```
Windows
>>>
```

PIRENS Institute of Business Management and Administration, Loni Bk.

Seat No: 4956 **Sign:**

Student Name: Vishal Shridar Vishwakarma

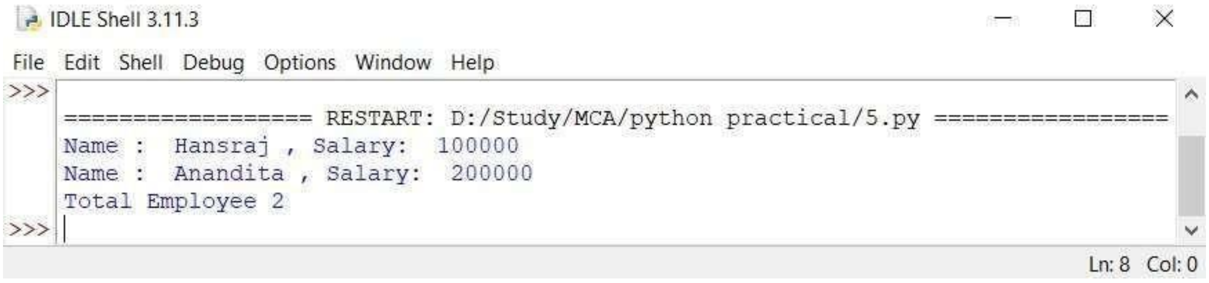
Subject Name: IT21L Practical

Program Title: 5) Programs for implementations of all object-oriented concepts like class, method, inheritance, polymorphism etc. (Real life examples must be covered for the implementation of object-oriented concepts)

Code:

```
class Employee:          empCount = 0
def___init_(self, name, salary):
          self.name = name
     self.salary = salary
          Employee.empCount += 1
     def displayCount(self):
          print("Total Employee %d" % Employee.empCount)
     def displayEmployee(self):
          print("Name : ", self.name, ", Salary: ", self.salary)
     empA = Employee("Hansraj",100000) empN =
     Employee("Anandita", 200000)
     empA.displayEmployee() empN.displayEmployee()
     print("Total Employee %d" % Employee.empCount)
```

Output:

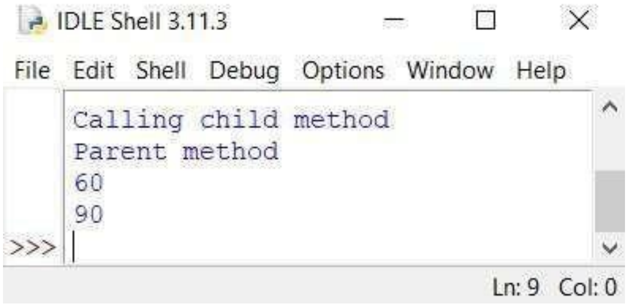


```
>>>
    ================== RESTART: D:/Study/MCA/python practical/5.py ==================
    Name :  Hansraj , Salary:  100000
    Name :  Anandita , Salary:  200000
    Total Employee 2
>>> |
```

2) Inheritance and Polymorphism

Code:
```
class Parent:           def
myMethod(self):
      print('Calling parent method')


   def parentMethod(self):
print("Parent method")


   def sum(self,x,y,z=20):
return (x+y+z)
class Child(Parent):
def myMethod(self):
      print('Calling child method')


c = Child()
c.myMethod()
c.parentMethod()      print(c.sum(20,20))
print(c.sum(20,20,50))
```

Output:



```
Calling child method
Parent method
60
90
>>>
```

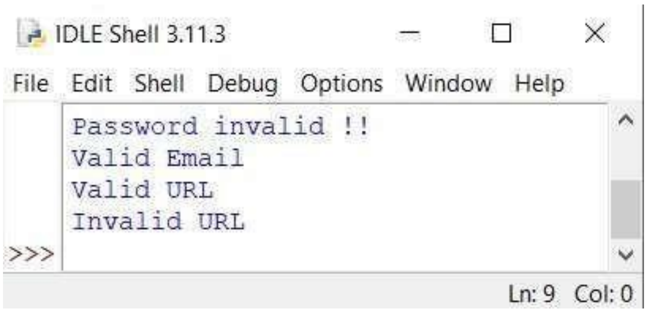
PIRENS Institute of Business Management and Administration, Loni Bk.	
Seat No: 4956	**Sign:**
Student Name: Vishal Shridar Vishwakarma	
Subject Name: IT21L Practical	
Program Title: 6) Programs for parsing of data, validations like Password, email, URL, etc	

Code: import re def

```
checkPassword(passwd):
   reg = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{6,20}$"
pat = re.compile(reg)     mat = re.search(pat, passwd)     if mat:
      print("Password is valid.")
else:

                    print("Password invalid !!")


def checkEmail(email):
   regex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
if (re.fullmatch(regex, email)):
      print("Valid Email")
else:
      print("Invalid Email")


def checkURL(url):
   rege = ("((http|https)://)(www.)?" +
          "[a-zA-Z0-9@:%._\\+~#?&//=]"  +
          "{2,256}\\.[a-z]" +
          "{2,6}\\b([-a-zA-Z0-9@:%" +
          "._\\+~#?&//=]*)")
p = re.compile(rege)
   if (re.search(p, url)):
```

```
print("Valid URL")     else:
        print("Invalid URL")


passw = "HGD@123" checkPassword(passw)


email = "support@gmail.com" checkEmail(email)


url = "https://www.ilovepillu.com" checkURL(url)


url2 = "www.pillu@test.com" checkURL(url2)
```

Output:



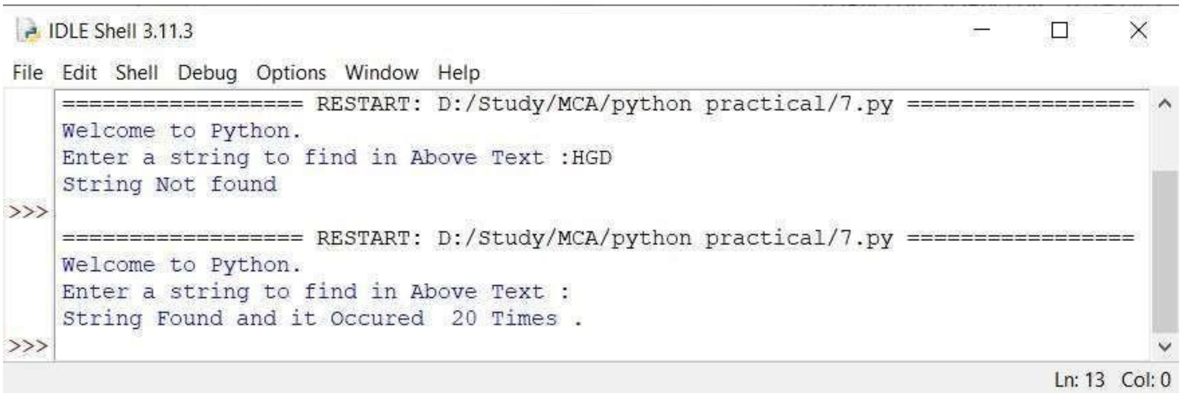
```
Password invalid !!
Valid Email
Valid URL
Invalid URL
>>>
```

PIRENS Institute of Business Management and Administration, Loni Bk.	
Seat No: 4956	**Sign:**
Student Name: Vishal Shridar Vishwakarma	
Subject Name: IT21L Practical	
Program Title: 7) Programs for Pattern finding should be covered.	

Code:

```
import re
txt = "Welcome to Python. " print(txt)
serchKey = input("Enter a string to find in Above Text :")
x = re.search(serchKey,txt) y=re.findall(serchKey,txt) if
x:     print("String Found and it Occured ", len(y), "Times
.") else:
   print("String Not found")
```

Output:



```
================== RESTART: D:/Study/MCA/python practical/7.py =================
Welcome to Python.
Enter a string to find in Above Text :HGD
String Not found
>>>
================== RESTART: D:/Study/MCA/python practical/7.py =================
Welcome to Python.
Enter a string to find in Above Text :
String Found and it Occured  20 Times .
>>>
```

PIRENS Institute of Business Management and Administration, Loni Bk.
Seat No: 4956 Sign:
Student Name: Vishal Shridar Vishwakarma
Subject Name: IT21L Practical
Program Title: 8) Programs covering all the aspects of Exception handling, user defined exception, Multithreading should be covered.

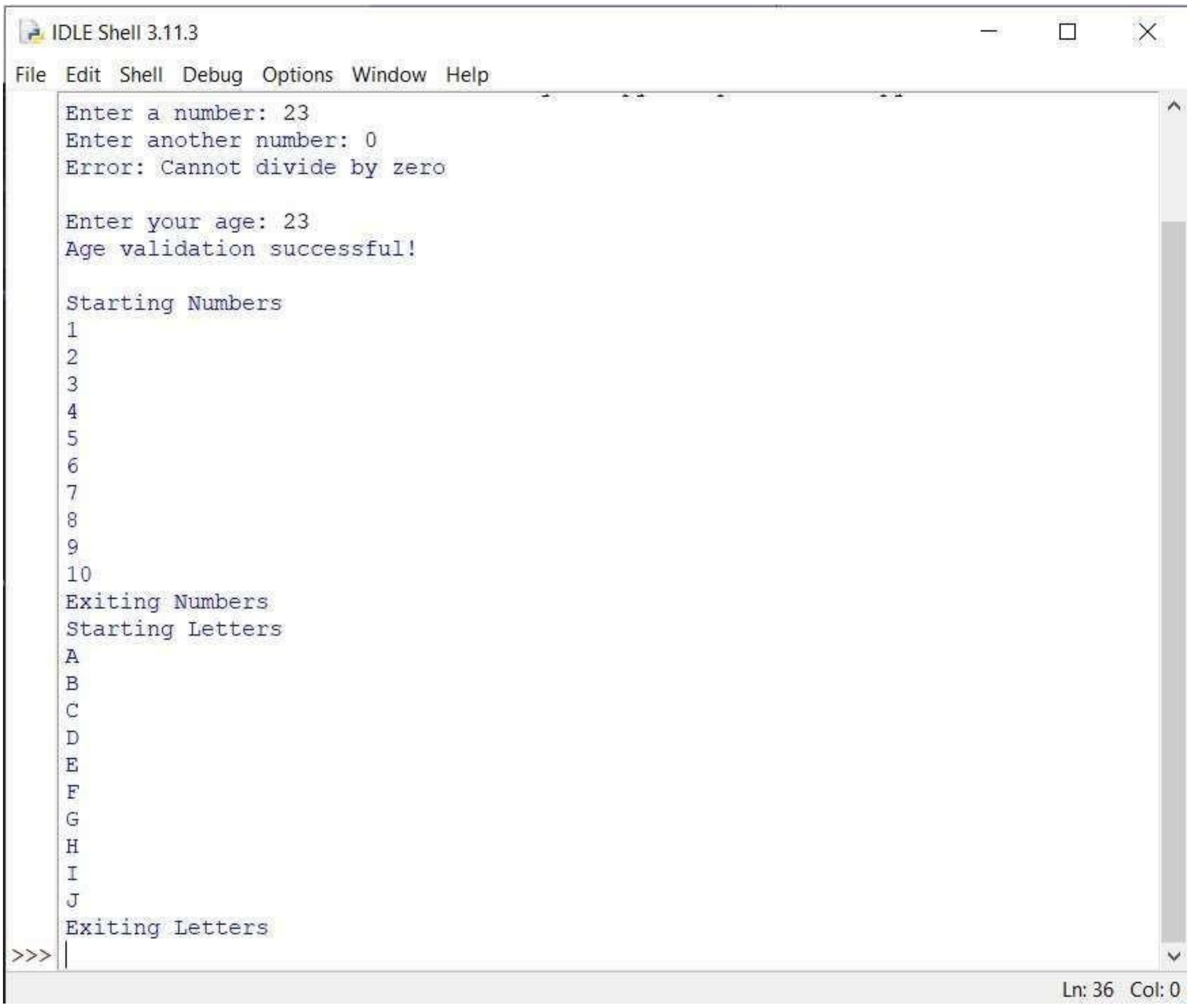
Code:

```
import threading class
DivideByZeroExp(Exception):
   pass def
divide_numbers(a, b):
try:       if b <= 0:
         raise DivideByZeroExp
div = a / b      print(f"Division
result: {div}")    except
DivideByZeroExp:
      print("Error: Cannot divide by zero")
except ZeroDivisionError:
      print("Error: Zero division error occurred")
except TypeError:
      print("Error: Invalid operand types")
except Exception as e:       print(f"Error:
{str(e)}") class
MyThread(threading.Thread):
   lock = threading.Lock()
def___init_(self, name):
super()._init__()
self.name = name    def
run(self):
    with self.lock:
      print(f"Starting {self.name}")
if self.name == "Numbers":
for i in range(1, 11):
          print(i)       elif
self.name == "Letters":         for
letter in "ABCDEFGHIJ":
          print(letter)
            print(f"Exiting {self.name}")
if _name___== "_main_":    #
Exception handling example   try:
    a = int(input("Enter a number: "))
b = int(input("Enter another number: "))
divide_numbers(a, b)   except ValueError:
    print("Error: Invalid input")
print()
  # User-defined exception example
try:
    age = int(input("Enter your age: "))
if age < 18:
      raise ValueError("Invalid age: Must be 18 or above to proceed.")
else:
      print("Age validation successful!")
except ValueError as e:    print(f"Error:
{str(e)}")   print()


  # Multithreading example
thread1 = MyThread("Numbers")
thread2 = MyThread("Letters")
```

thread1.start()   thread2.start()

thread1.join()   thread2.join()

Output:



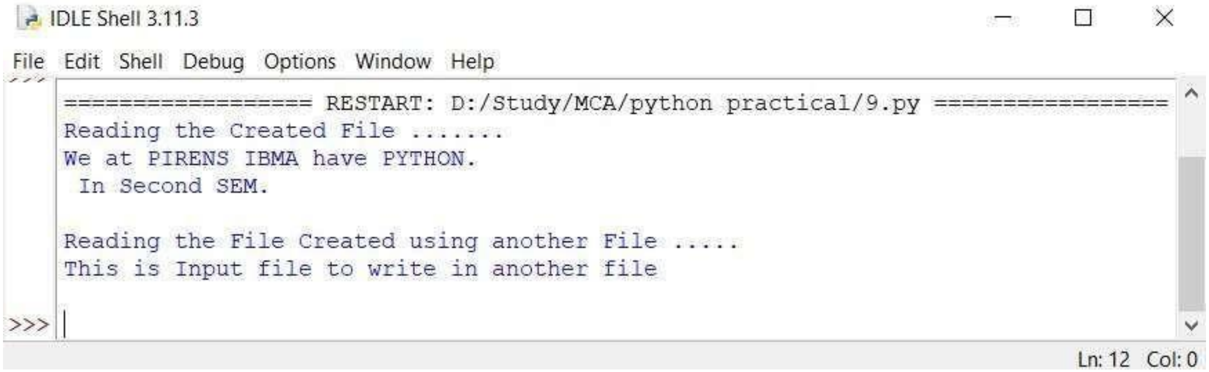
```
Enter a number: 23
Enter another number: 0
Error: Cannot divide by zero

Enter your age: 23
Age validation successful!

Starting Numbers
1
2
3
4
5
6
7
8
9
10
Exiting Numbers
Starting Letters
A
B
C
D
E
F
G
H
I
J
Exiting Letters
>>>
```

PIRENS Institute of Business Management and Administration, Loni Bk.
Seat No: 4956 Sign:
Student Name: Vishal Shridar Vishwakarma
Subject Name: IT21L Practical
Program Title: 9) Programs demonstrating the IO operations like reading from file, writing into file from different file types like data file, binary file, etc.

Code:

```
fo = open("Python.txt", "w")

fo.write("We at PIRENS IBMA have PYTHON.\n In Second SEM. \n") fo.close()
output_f = open("Python.txt", "r+")

print("Reading the Created File .......")

print(output_f.read()) output_f.close()


fo2 = open("Input_File.txt", "w")

fo2.write("This is Input file to write in another file\n") fo2.close()

input_file = open("Input_File.txt", "r+")

output_file = open("Output_Python.txt","w") with

input_file as scan:

 output_file.write(scan.read()) output_file.close()

out_f = open("Output_Python.txt", "r+")

print("Reading the File Created using another File......")

print(out_f.read()) out_f.close()
```

Output:



```
================== RESTART: D:/Study/MCA/python practical/9.py =================
Reading the Created File .......
We at PIRENS IBMA have PYTHON.
 In Second SEM.

Reading the File Created using another File .....
This is Input file to write in another file

>>> |
```

PIRENS Institute of Business Management and Administration, Loni Bk.

Seat No: 4956 **Sign:**

Student Name: Vishal Shridar Vishwakarma

Subject Name: IT21L Practical

Program Title:10) Programs to perform searching, adding, updating the content from the file.

Code:

```
read_F = open("Python.txt","r+")
print(read_F.read()) string1 =
'Python' file1 =
open("Python.txt", "r+") flag = 0
index = 0
# Loop through the file line by line
for line in file1:     index += 1
    if string1 in line:
        flag = 1
break
print("Searching ...")
if flag == 0:
    print('String', string1, 'Not Found') else:
    print('String', string1, 'Found In Line', index)
file1.close() print("Updating ...") reading_file
= open("Python.txt", "r+") new_file_content =
"" for line in reading_file: stripped_line =
line.strip()
  new_line = stripped_line.replace("Python", "Java")
new_file_content += new_line +"\n" reading_file.close()
writing_file = open("Python.txt", "w")
writing_file.write(new_file_content) writing_file.close()
```

writing_file2 = open("Python.txt", "r+")

print(writing_file2.read())

writing_file2.close() print("Adding ...")

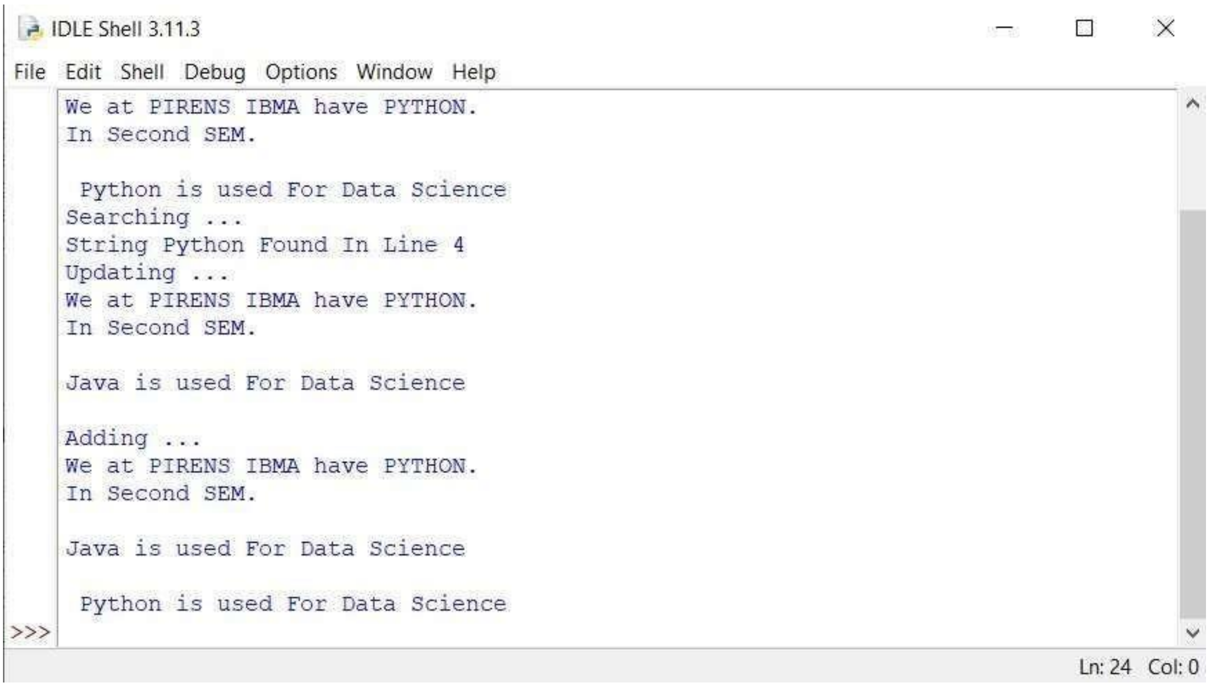
appending_file = open("Python.txt","a")

appending_file.write("\n Python is used For Data Science")

appending_file.close() append_F=open("Python.txt","r+")

print(append_F.read())

Output:



```
We at PIRENS IBMA have PYTHON.
In Second SEM.

 Python is used For Data Science
Searching ...
String Python Found In Line 4
Updating ...
We at PIRENS IBMA have PYTHON.
In Second SEM.

Java is used For Data Science

Adding ...
We at PIRENS IBMA have PYTHON.
In Second SEM.

Java is used For Data Science

 Python is used For Data Science
>>>
```

PIRENS Institute of Business Management and Administration, Loni Bk.
Seat No: 4956 Sign:
Student Name: Vishal Shridar Vishwakarma
Subject Name: IT21L Practical
Program Title:11) Program for performing CRUD operation with MongoDB and.

Code:

```
import pymongo
        connection_url="mongodb://localhost:81/"
client=pymongo.MongoClient(connection_url)


database_name="student_database"  student_db=client[database_name]


collection_name="computer science" collection=student_db[collection_name]


documents=[{"Name":"Akshay","Roll No":01,"Branch":"MCA"},{"Name":"Nitin","Roll
No":02,"Branch":"MCA"},{"Name":"Shivam","Roll No":03,"Branch":"MCA"}]
collection.insert_many(documents) print("Inserted Data : ")
query={"Branch":"MCA"}
result=collection.find(query) for
i in result:
    print(i)
query2={"Roll No":{"$eq":03}} present_data=collection.find_one(query2)
new_data={"$set":{"Name":'Krushna'}}
collection.update_one(present_data,new_data)


print("After Updation : ")
query={"Branch":"MCA"}
result=collection.find(query) for
i in result:
    print(i)
```

```
query={"Roll No":02} collection.delete_one(query)


print("After Deletion : ")
query={"Branch":"MCA"}
result=collection.find(query) for
i in result:
    print(i)
```

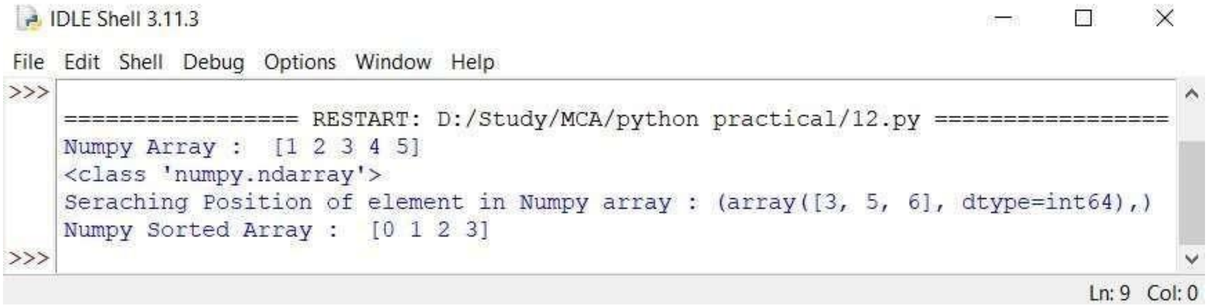
```
/*OUTPUT*/
Inserted Data :
{'_id': ObjectId('396d3870bc5c552'), 'Name': 'Akshay', 'Roll No': 01, 'Branch': 'MCA'}
{'_id': ObjectId('396d3870bc5c553'), 'Name': 'Nitin', 'Roll No': 02, 'Branch': 'MCA'}
{'_id': ObjectId('396d3870bc5c554'), 'Name': 'Shivam', 'Roll No': 03, 'Branch': 'MCA'}
After Updation :
{'_id': ObjectId('396d3870bc5c552'), 'Name': 'Akshay', 'Roll No': 01, 'Branch': 'MCA'}
{'_id': ObjectId('396d3870bc5c553'), 'Name': 'Nitin'', 'Roll No': 02, 'Branch': 'MCA'}
{'_id': ObjectId('396d3870bc5c554'), 'Name': 'Krushna', 'Roll No': 03, 'Branch': 'MCA'}
After Deletion :
{'_id': ObjectId('396d3870bc5c552'), 'Name': 'Akshay', 'Roll No': 01, 'Branch': 'MCA'}
{'_id': ObjectId('396d3870bc5c553'), 'Name': 'Nitin', 'Roll No': 02, 'Branch': 'MCA'}
```

PIRENS Institute of Business Management and Administration, Loni Bk.
Seat No: 4956 Sign:
Student Name: Vishal Shridar Vishwakarma
Subject Name: IT21L Practical
Program Title:12) Basic programs with NumPy as Array, Searching and Sorting, date & time and String handling.

Code: import numpy as

```
np

#numpy    array    arr    =
np.array([1,2,3,4,5])
print("Numpy Array : ",arr)
print(type(arr))

#numpy array search
arr2 = np.array([1, 2, 3, 4, 5, 4, 4]) x
= np.where(arr2 == 4)
print("Seraching Position of element in Numpy array :",x)

#numpy sorting arr3 =
np.array([3, 2, 0, 1])
print("Numpy Sorted Array : ",np.sort(arr3))
```

Output:



```
>>>
================= RESTART: D:/Study/MCA/python practical/12.py =================
Numpy Array :  [1 2 3 4 5]
<class 'numpy.ndarray'>
Seraching Position of element in Numpy array : (array([3, 5, 6], dtype=int64),)
Numpy Sorted Array :  [0 1 2 3]
>>>
```

2) Write program with numPy to handle string in python.

Code:

```
import numpy as np #

converting to lowercase
```

print(np.char.lower(['PYTHON', 'JAVA']))

# converting to lowercase print(np.char.lower('PYTHON'))

# splitting a string

print(np.char.split('Python, Java, PHP', sep = ','))

# splitting a string

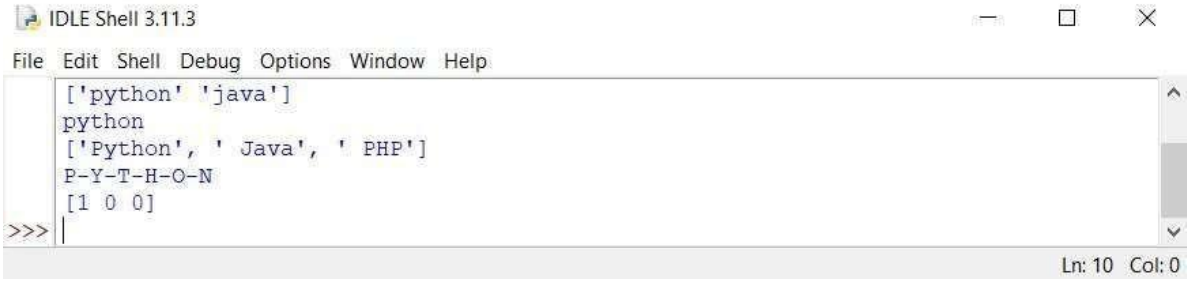
print(np.char.join('-', 'PYTHON')) a

= np.array(['python', 'java', 'php'])

# counting a substring print(np.char.count(a,

'python'))

Output:



```
['python' 'java']
python
['Python', ' Java', ' PHP']
P-Y-T-H-O-N
[1 0 0]
>>>
```

PIRENS Institute of Business Management and Administration, Loni Bk.

Seat No: 4956 **Sign:**

Student Name: Vishal Shridar Vishwakarma

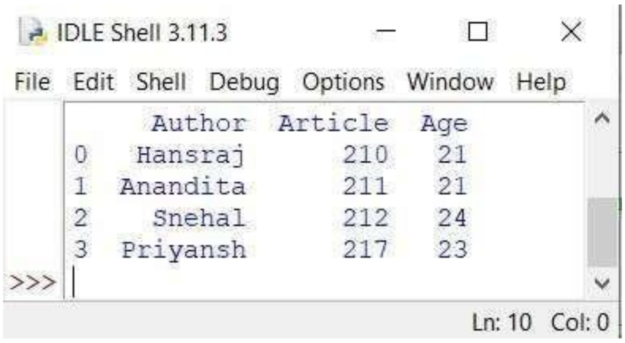
Subject Name: IT21L Practical

Program Title: 13) Programs for series and data frames should be covered.

Code:

```
import pandas as pd author = ['Akshay',
'Nitin', 'Ram', 'Sham'] article = [210, 211,
212, 217] auth_series =
pd.Series(author) article_series =
pd.Series(article)
frame = { 'Author': auth_series, 'Article': article_series }
result = pd.DataFrame(frame) age = [21, 21, 24, 23]
result['Age'] = pd.Series(age) print(result)
```

Output:



```
     Author  Article  Age
0   Hansraj      210   21
1  Anandita      211   21
2    Snehal      212   24
3  Priyansh      217   23
>>> 
```

PIRENS Institute of Business Management and Administration, Loni Bk.
Seat No: 4956 Sign:
Student Name: Vishal Shridar Vishwakarma
Subject Name: IT21L Practical
Program Title:14) Programs to demonstrate data pre-processing and data handling with data frame.

Code:

```
# importing libraries
import pandas
import scipy import
numpy
from sklearn.preprocessing import MinMaxScaler
 # data set link
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-
diabetes/pimaindians-diabetes.data" # data parameters
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
 # preparating of dataframe using the data at given link and defined columns list
dataframe = pandas.read_csv(url, names = names) array = dataframe.values
 # separate array into input and output components
X = array[:,0:8]
Y = array[:,8]
 # initialising the MinMaxScaler scaler =
MinMaxScaler(feature_range=(0, 1))
# learning the statistical parameters for each of the data and transforming
rescaledX = scaler.fit_transform(X) # summarize transformed data
numpy.set_printoptions(precision=3) print(rescaledX[0:5,:])
```

Output:

```
 [[ 0.353   0.744   0.59                  0.354   0.0             0.501   0.234   0.483]
 [ 0.059         0.427       0.541      0.293       0.0            0.396     0.117      0.167]
 [ 0.471         0.92         0.525      0.             0.0            0.347     0.254      0.183]   [
0.059   0.447   0.541   0.232   0.111   0.419   0.038   0.0   ]
 [ 0.0             0.688   0.328   0.354   0.199   0.642   0.944   0.2   ]]
```

PIRENS Institute of Business Management and Administration, Loni Bk.	
Seat No: 4956	**Sign:**
Student Name: Vishal Shridar Vishwakarma	
Subject Name: IT21L Practical	
Program Title:15) Program for data visualization should be covered	

Code:

```
import seaborn as sns import
matplotlib.pyplot as plt # Load the
example dataset from Seaborn
titanic_dataset = sns.load_dataset("titanic")
# Creating bar plot
sns.barplot(x='fare', y='who', data=titanic_dataset, palette="Blues")
# Adding the aesthetics
plt.title('Chart title') plt.xlabel('X axis title')plt.ylabel('Y axis title')# Show the plot plt.show()
```

Output:

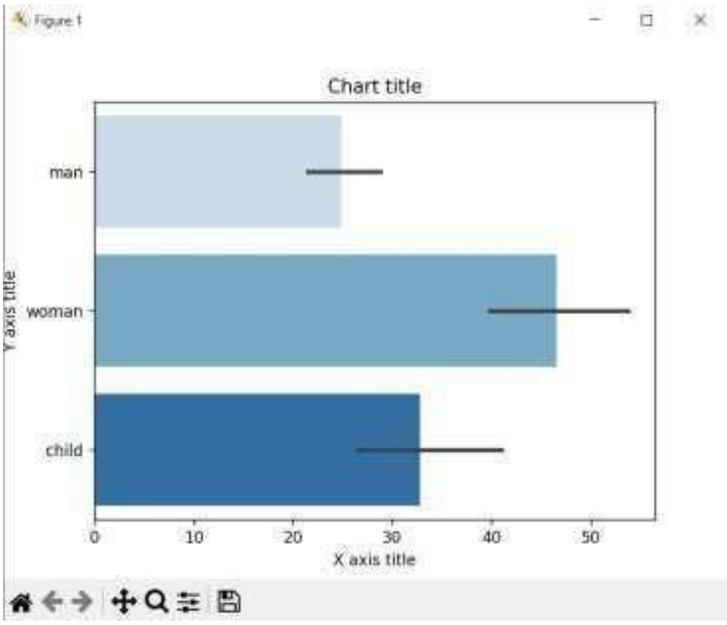


Figure 1
Chart title
man
woman
child
Y axis title
0
10
20
30
40
50
X axis title