

PYTHON LAB MANUAL

Designed for

UNIVERSITY OF MUMBAI



University of Mumbai

As per REVISED SYLLABUS (R-2016)
OF INFORMATION TECHNOLOGY
ENGINEERING

Subject Experts: - Dr. PROF. VAISHALI KHAIRNAR

PROF. BHUSHAN JADHAV

SYLLABUS

Course Code	Course Name	Theory	Practical	Tutorial	Theory	TW/Practical	Tutorial	Total
ITL404	Python lab	--	2+2*	--	--	02	--	02

Course Code	Course Name	Examination Scheme									
		Theory Marks				Term Work	Oral & Practical	Total			
		Internal assessment									
		Test1	Test 2	Avg. of two Tests	End Sem. Exam						
ITL404	Python lab	--	--	--	--	50	50	100			

*2 hours shown as practical's to be taken class wise lecture and other 2 hours to be taken as batch wise practicals in Lab.

Lab Objectives: The course will help the students to get familiar with:

1. Basics of Python programming
2. Decision Making and Functions in Python
3. Object Oriented Programming using Python
4. Files Handling in Python
5. GUI Programming and Databases operations in Python
6. Network Programming in Python

Lab Outcomes: Upon Completion of the course the learner should be able to:

1. Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python
2. Express different Decision Making statements and Functions
3. Interpret Object oriented programming in Python
4. Understand and summarize different File handling operations
5. Explain how to design GUI Applications in Python and evaluate different database operations
6. Design and develop Client Server network applications using Python

Hardware & Software Requirements:

Hardware Requirements	Software Requirements	Other Requirements
PC With following Configuration	1. Windows or Linux Desktop OS 2. Python 3.6 or higher	1. Internet Connection for installing additional packages

1. Intel PIV Processor 2. 2 GB RAM 3. 500 GB Harddisk 4. Network interface card	3. Notepad ++ 4. Python IDEs like Pydev, Netbeans or Eclipse 5. Mysql	
--	---	--

Prerequisite Subjects: Structured Programming Approach & Java Programming

Detailed Syllabus:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Basic Programming syntax of Java/C. Installation and configuration of python.	02	
I	Basics of Python	<p>Theory: Numbers in Python, Basic & Built-in Math functions, Number Formats, Strings, Quotes, print() Function, Assigning Values to Names & Changing Data Through Names, Copying Data, Tuples — Unchanging Sequences of Data, Lists — Changeable Sequences of Data, Dictionaries — Groupings of Data Indexed by Name, Special String Substitution Using Dictionaries , Arrays, Treating a String Like a List, Special Types, Ranges of Sequences, Working with Sets, Arrays.</p> <p>Lab Experiment:</p> <p>Write python programs to understand Expressions, Variables, Quotes, Basic Math operations, Strings: Basic String Operations & String Methods, List, Tuples, Dictionaries, Arrays.</p>	10	LO 1

		(Minimum Three Programs based on math operations, Strings and List/Tuples/ Dictionaries)		
II	Decision Making and Functions	<p>Theory: If statement, if-elif-else, Repetition using while loop, for loop, break statement, Handling Errors- try: statement, except: statement, Functions-Grouping Code under a Name, defining a Function, describing a function in the function, Checking & Setting Your Parameters, Calling Functions from within Other Functions, Functions Inside of Functions, Layers of Functions</p> <p>Lab Experiment:</p> <p>Write python programs to understand different decision making statements and Functions.</p> <p>(Minimum Three Programs based on Decision making, Looping Statements and Functions)</p>	08	LO 2
III	Object Oriented Programming	<p>Theory: Creating a Class, Self-Variables, Constructors, Types of Methods, Inner Classes, Constructors in Inheritance, Polymorphism,, The super() Method, Method Resolution Order (MRO), Operator Overloading, Method Overloading & Overriding, Interfaces in Python. Exceptions Handling: Errors in a Python Program, Exceptions, Exception Handling, Types of Exceptions, The Except Block, The assert Statement.</p>	08	LO 3

		<p>Modules and Packages: Creating Modules and Packages, Documenting & Viewing Module, Basics of Testing Your Modules and Packages, Importing & exporting Modules.</p> <p>Lab Experiment:</p> <p>Write python programs to understand different Object oriented features in Python</p> <p>(Minimum four programs based on</p> <ul style="list-style-type: none"> a) Classes & objects, b) Constructors, c) Inheritance & Polymorphism, d) Exception handling 		
IV	Files Handling	<p>Theory: Types of Files in Python, Opening a File, Closing a File. Writing Text Files, Knowing Whether a File Exists or Not, Working with Binary Files, Appending Text to a File, Reading Text Files, File Exceptions, The with Statement</p> <p>Pickle in Python, Lambda and Filter, Map & range functions.</p> <p>Lab Experiment:</p> <p>Write python programs to understand different File handling operations</p>	06	LO 4
V	GUI Programming and Databases	<p>Theory: GUI Programming - Writing a GUI with Python: GUI Programming Toolkits, Creating GUI Widgets with Tkinter, Creating</p>	08	LO 5

		<p>Layouts, Radio Buttons and Checkboxes, Dialog Boxes.</p> <p>Database Access - Python's Database Connectivity, Types of Databases Used with Python, Mysql database Connectivity with Python, Performing Insert, Deleting & Update operations on database</p> <p>Lab Experiment:</p> <p>Write python programs to understand GUI designing and database operations</p> <p>(Minimum Three programs based on GUI designing using Tkinter, Mysql database creation & Database connectivity with DML operations using python)</p>		
VI	Network Programming	<p>Theory: Understanding Protocols, Introduction to Sockets, TCP/IP Server, TCP/IP Client, UDP Server, UDP Client, File Server, File Client, Two-Way Communication between Server and Client, Multithreaded Client-Server Chat Application</p> <p>Lab Experiment:</p> <p>Write python programs to understand TCP and UDP Sockets in Python</p> <p>(Minimum One programs based on TCP or UDP Sockets)</p>	06	LO 6

Text Books:

1. James Payne, "Beginning Python: Using Python 2.6 and Python 3.1", Wrox Publication
2. Dr. R. Nageswara Rao,"Core Python Programming" , Dreamtech Press, Wiley Publication.
3. Magnus Lie Hetland,"Beginning Python From Novice to Professional", Second Edition", Apress Publication.

Reference Books:

1. Wesley J Chun," Core Python Applications Programming",Third Edition, Pearson Publication.
2. E. Balguruswamy," Introduction to Computing and Problem Solving using Python", McGraw Hill Publication
3. David Beazley,"Python Essential Reference", Sams Publishing

Term Work:

Term Work shall consist of at least 12 to 15 practical's based on the below list. Also Term work Journal must include at least 2 assignments.

Term Work Marks: 50 Marks (Total marks) = 40 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Oral & Practical Exam: An Oral & Practical exam will be held based on the above syllabus.

LIST OF EXPERIMENTS

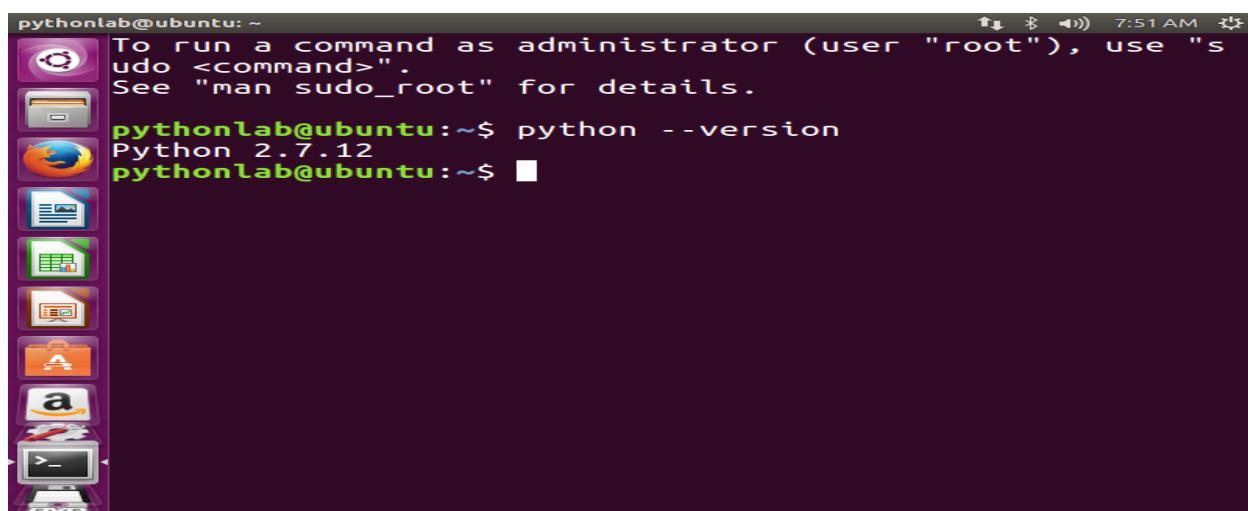
Sr. No.	Name of Experiment	LO
1	<p>1.1 Write a python program to implement Comments, Datatypes, Expressions, Input and Output Functions</p> <p>1.2 Write a python program to implement Byte Array, Range, Set and STRING Functions</p> <p>1.3 Write a python program to implement List, Tuple, Dictionaries and Arrays</p>	LO 1
2	<p>2.1 Write python programs to implement the Control Structures</p> <p>2.2 Write python programs to implement the For Loop</p> <p>2.3 Write python programs to implement Functions</p>	LO 2
3	<p>3.1 Write python programs to implement Classes, object, Static method and inner class</p> <p>3.2 Write a python program to implement Constructors</p> <p>3.3 Write python programs to implement Inheritance and Polymorphism with Method overloading and Method Overriding</p> <p>3.4 Write python program to implement different types of Exceptions</p>	LO 3
4	<p>4.1 Write a python programs to implement different file handling operations using pickle</p> <p>4.2 Write a python program to implement Lambda, map, reduce, filter and range functions</p>	LO 4
5	<p>5.1 Write a python program to implement GUI Canvas Application using Tkinter</p> <p>5.2 Write a python program to implement GUI Frame Application using Tkinter</p>	LO 5

	5.3 Write a python program to implement CRUD Operations using Mysql Python Database Connectivity	
6	6.1 Write a python program to implement Client-Server programming using TCP Socket 6.2 Write a python program to implement Client-Server programming using UDP Socket	LO 6

STEPS TO INSTALL PYTHON RUNTIME ENVIRONMENT AND IDES

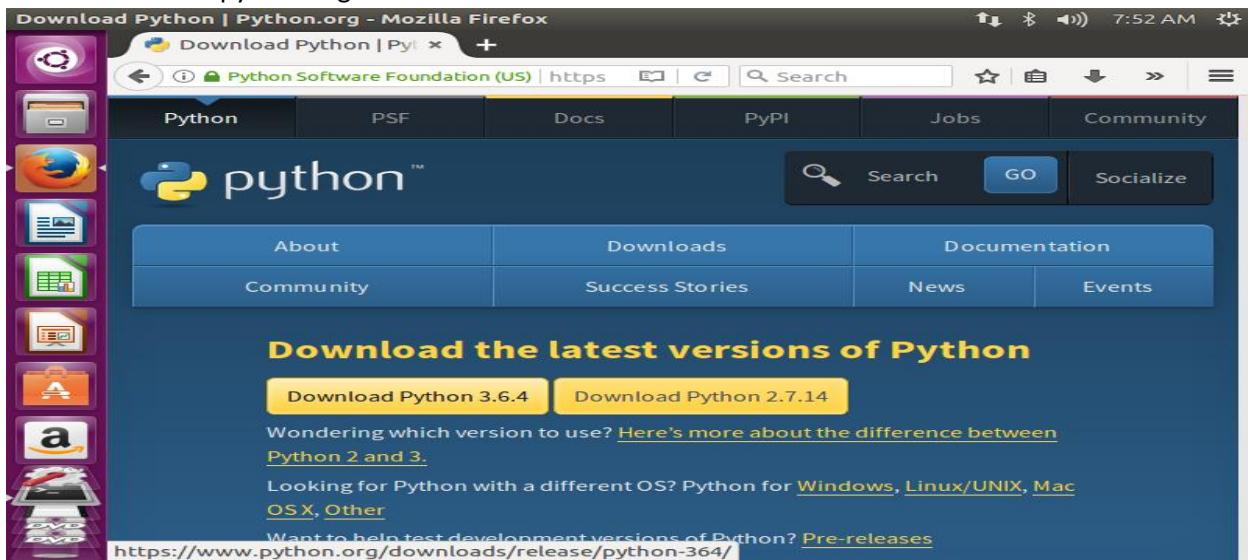
Ubuntu 14.04 and above versions has built-in Python but every new version of python comes with many new features and as we are going to use IDEs like Pydev and Pycharm, they require latest versions of it.

To check current Version of python run following command.



```
pythonlab@ubuntu:~$ To run a command as administrator (user "root"), use "sudo <command>". See "man sudo_root" for details.
pythonlab@ubuntu:~$ python --version
Python 2.7.12
pythonlab@ubuntu:~$
```

As we can see Ubuntu has preinstalled python version 2.7.12 but we want its latest version 3.6.4 so download it from python.org.



The python 3.6.4 for Ubuntu comes with compressed file tgz. So uncompressed it using tar command

```
$tar -xvf Python-3.6.4.tgz
```

Then go to python directory

```
$cd Python-3.6.4
```

Execute following commands in a sequence to complete the installation

```
$apt-get install zlib1g-dev  
$./configure  
$make  
$make install
```

Perform the installation, check the version and run the program 1.py.

```
pythonlab@ubuntu: ~/Downloads  
pythonlab@ubuntu:~$ cd Downloads/  
pythonlab@ubuntu:~/Downloads$ ls  
Python-3.6.4.tgz  
pythonlab@ubuntu:~/Downloads$ tar -xvf Python-3.6.4.tgz  
z  
Python-3.6.4/Objects/clinic/  
Python-3.6.4/Objects/clinic/unicodeobject.c.h  
Python-3.6.4/Objects/clinic/bytearrayobject.c.h  
Python-3.6.4/Objects/clinic/bytesobject.c.h  
Python-3.6.4/Objects/clinic/dictobject.c.h  
Python-3.6.4/Objects/bytarrayobject.c  
Python-3.6.4/Objects/typeobject.c  
Python-3.6.4/Objects/lnotab_notes.txt  
Python-3.6.4/Objects/methodobject.c  
Python-3.6.4/Objects/tupleobject.c  
Python-3.6.4/Objects/obmalloc.c  
Python-3.6.4/Objects/object.c  
Python-3.6.4/Objects/abstract.c  
Python-3.6.4/Objects/listobject.c  
Python-3.6.4/Objects/bytess_methods.c  
Python-3.6.4/Objects/dictnotes.txt  
Python-3.6.4/Objects/typeslots.inc  
pythonlab@ubuntu:~/Downloads$  
pythonlab@ubuntu:~/Downloads$ ./configure  
pythonlab@ubuntu:~/Downloads$ checking build system type... x86_64-pc-linux-gnu  
pythonlab@ubuntu:~/Downloads$ checking host system type... x86_64-pc-linux-gnu  
pythonlab@ubuntu:~/Downloads$ checking for python3.6... no  
pythonlab@ubuntu:~/Downloads$ checking for python3... python3  
pythonlab@ubuntu:~/Downloads$ checking --enable-universalsdk... no  
pythonlab@ubuntu:~/Downloads$ checking --with-universal-archs... no  
pythonlab@ubuntu:~/Downloads$ checking MACHDEP... linux  
pythonlab@ubuntu:~/Downloads$ checking for --without-gcc... no  
pythonlab@ubuntu:~/Downloads$ checking for --with-icc... no  
pythonlab@ubuntu:~/Downloads$ checking for gcc... gcc  
pythonlab@ubuntu:~/Downloads$ checking whether the C compiler works...  
pythonlab@ubuntu:~/Downloads$ sudo make install  
[sudo] password for pythonlab:
```

```

pythonlab@ubuntu:~/Downloads/Python-3.6.4$ python3 --version
Python 3.6.4
pythonlab@ubuntu:~/Downloads/Python-3.6.4$ 

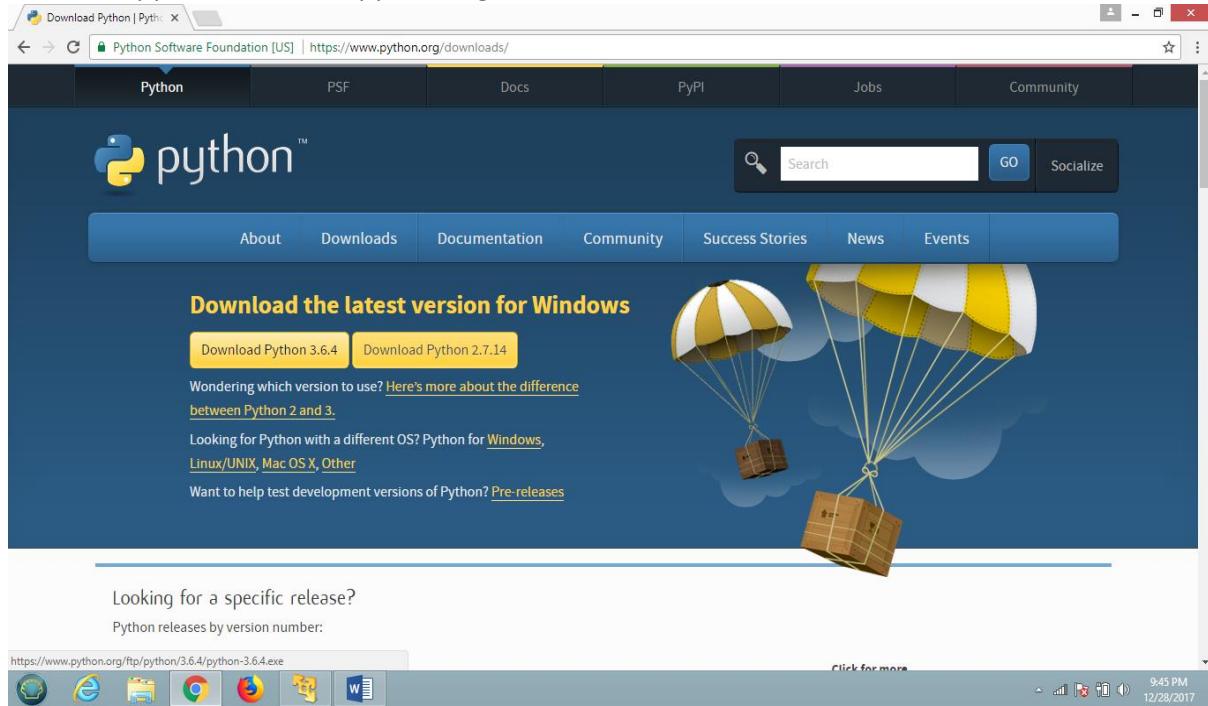
Terminal 1
#!/usr/bin/python
print("Enter value of a")
a=input()
print("Enter value of b")
b=input()
c=a+b
print('The sum of {} and {} is {}'.format(a, b, c))

Terminal 2
pythonlab@ubuntu:~$ python 1.py
Enter value of a
12
Enter value of b
23
The sum of 12 and 23 is 35
pythonlab@ubuntu:~$ 

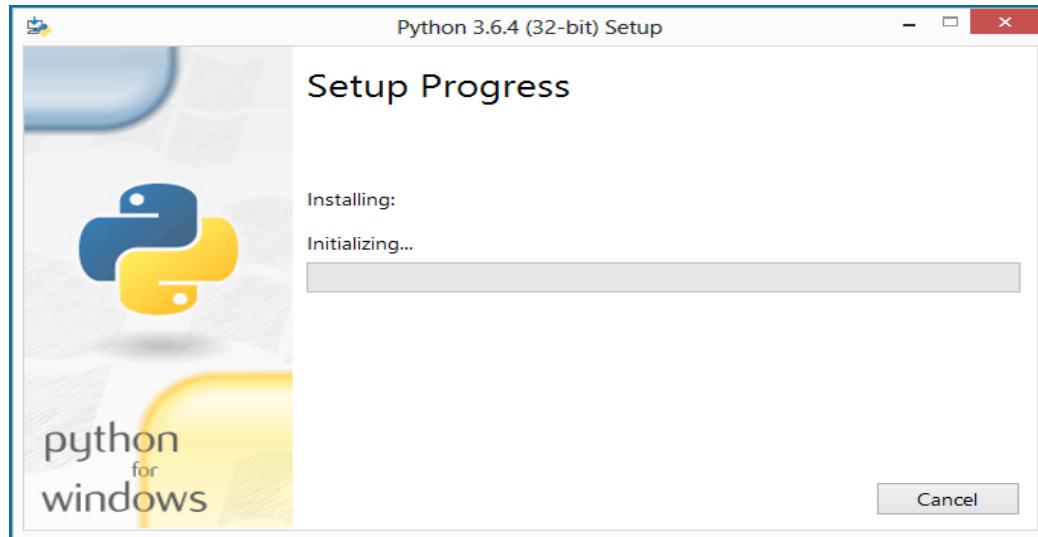
```

Installation of Python on windows

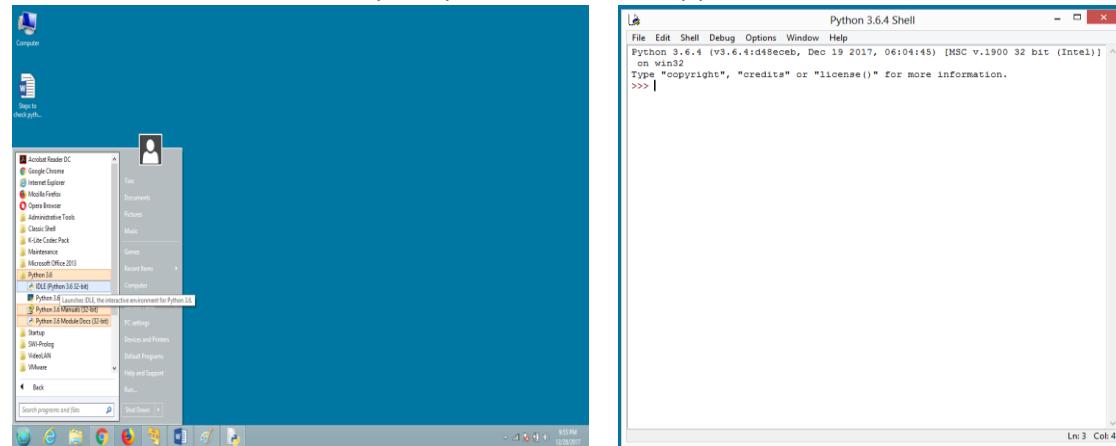
- 1) Download python 3.6.4 from python.org



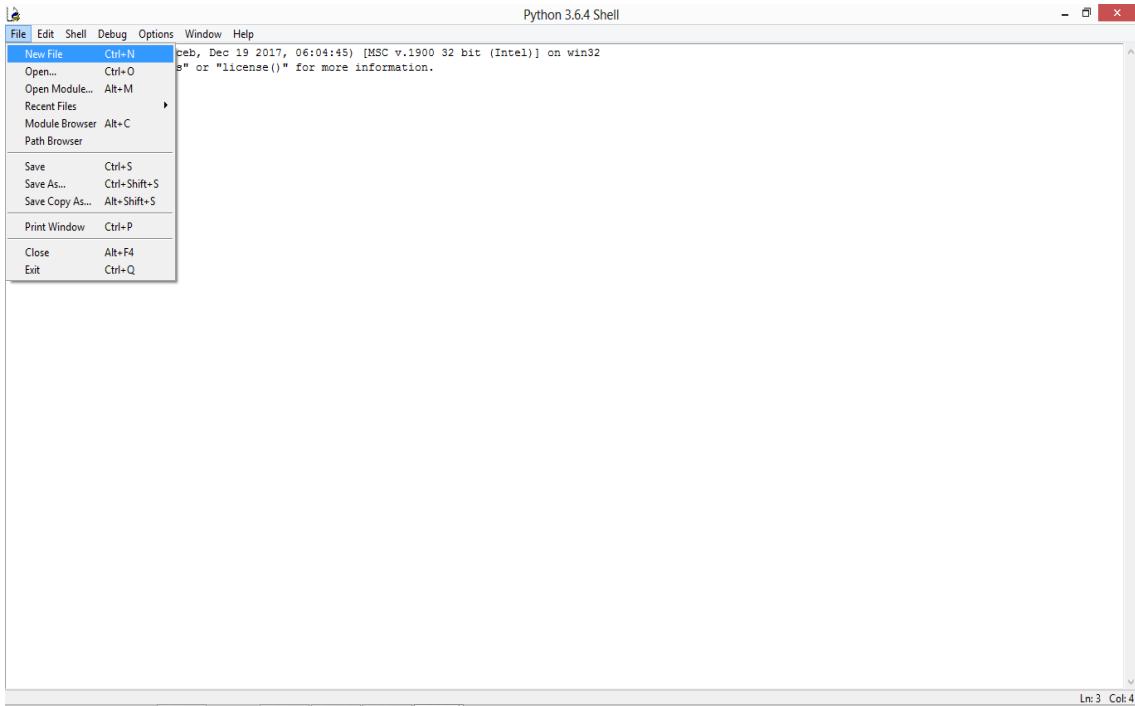
- 2) Install it by answering simple questions



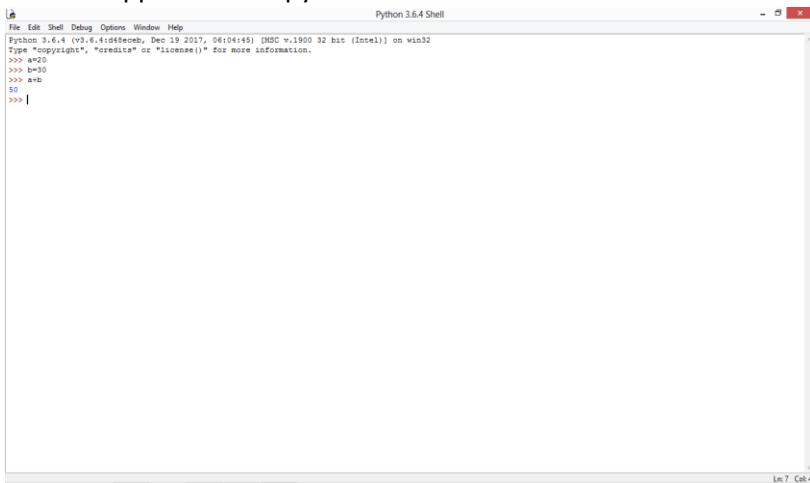
- 3) Once installation is done then open Python shell or IDLE (python 3.6.4 32 bit) from start menu



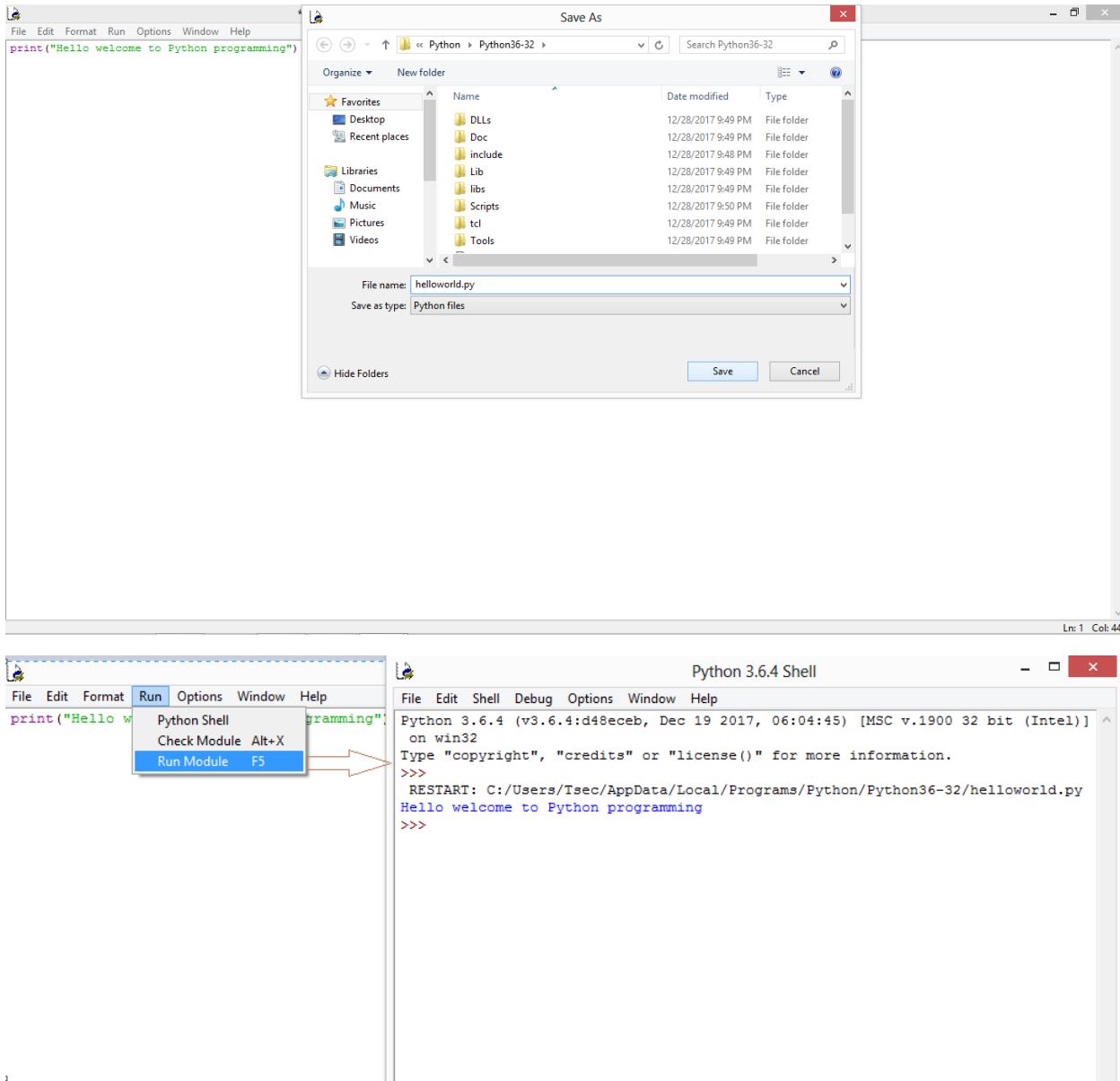
IDLE



4) Test the application on python shell



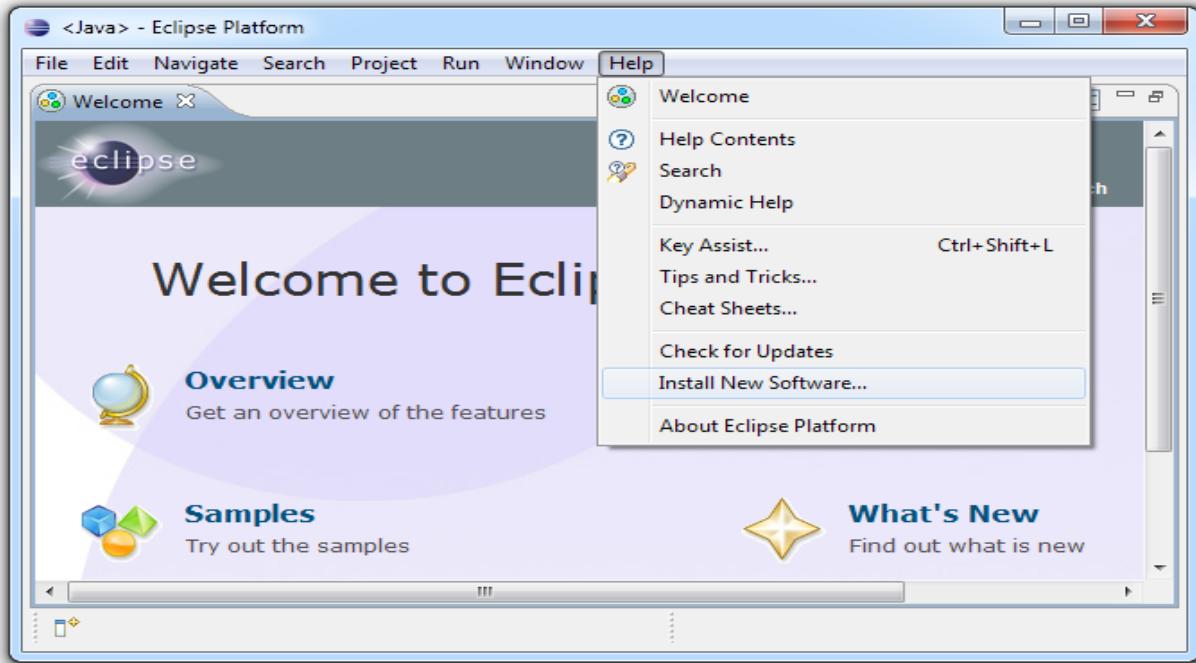
On IDLE



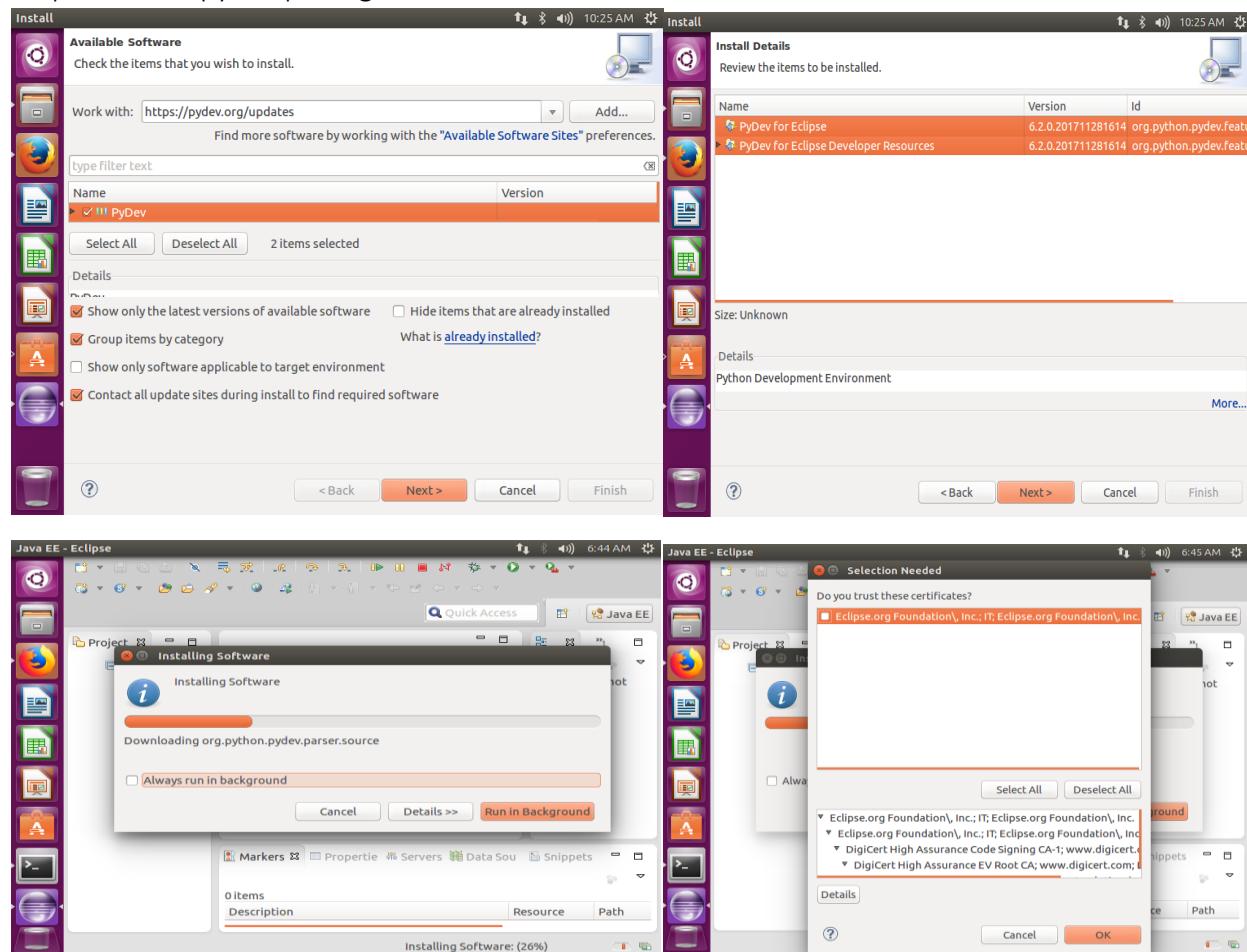
STEPS TO INSTALL PYDEV IN ECLIPSE

Step 1:- Install Eclipse in Ubuntu through Software center and in windows by downloading it through google

Step 2 :- Open Eclipse → Go to install new software under help menu and type <https://pydev.org/updates> in the search box → click on add button → specify name as Pydev and select Pydev

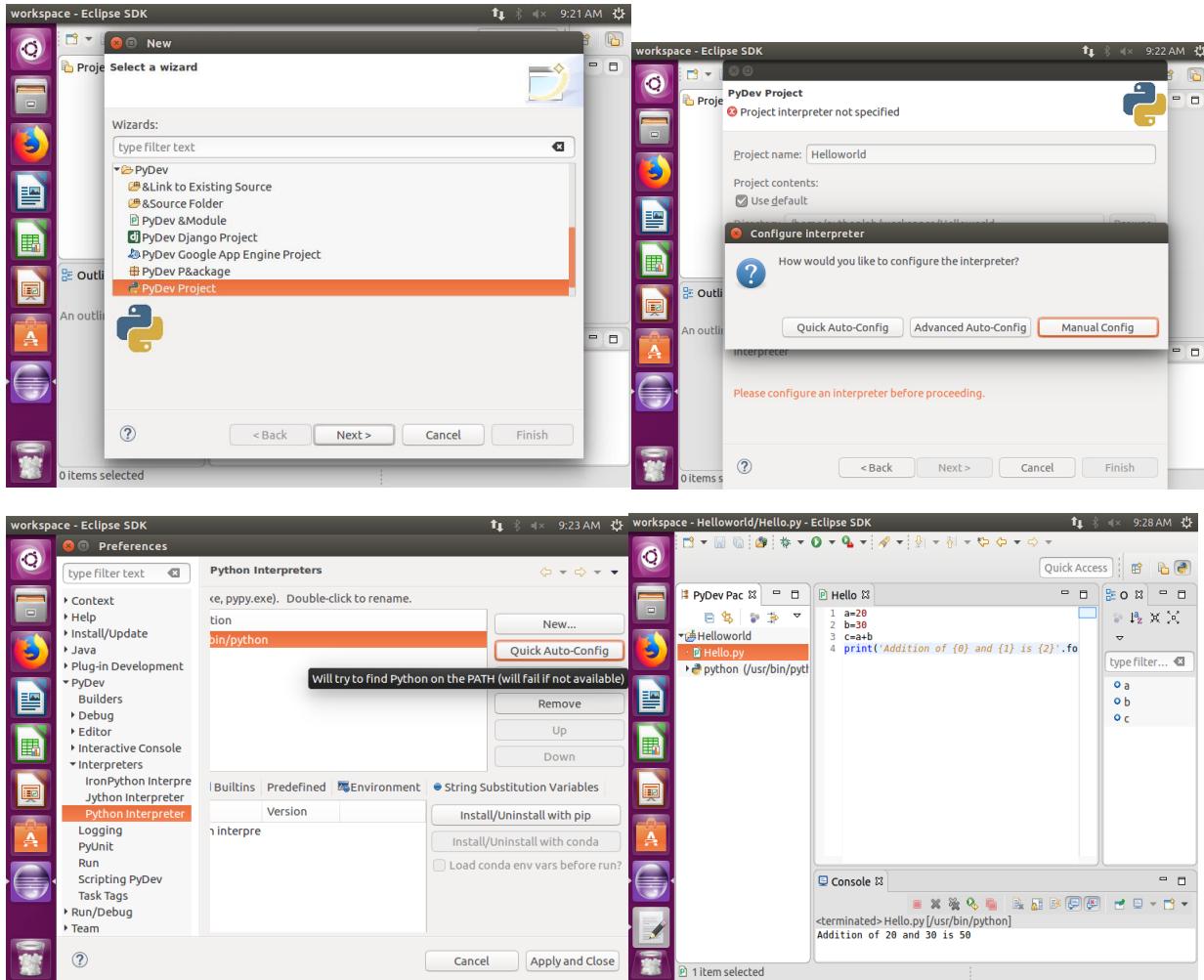


Step 3:- select pydev packages and click on next to finish the installation



Restart the eclipse when installation is done.

Step 4 :- Set the path to the interpreter and test the application.



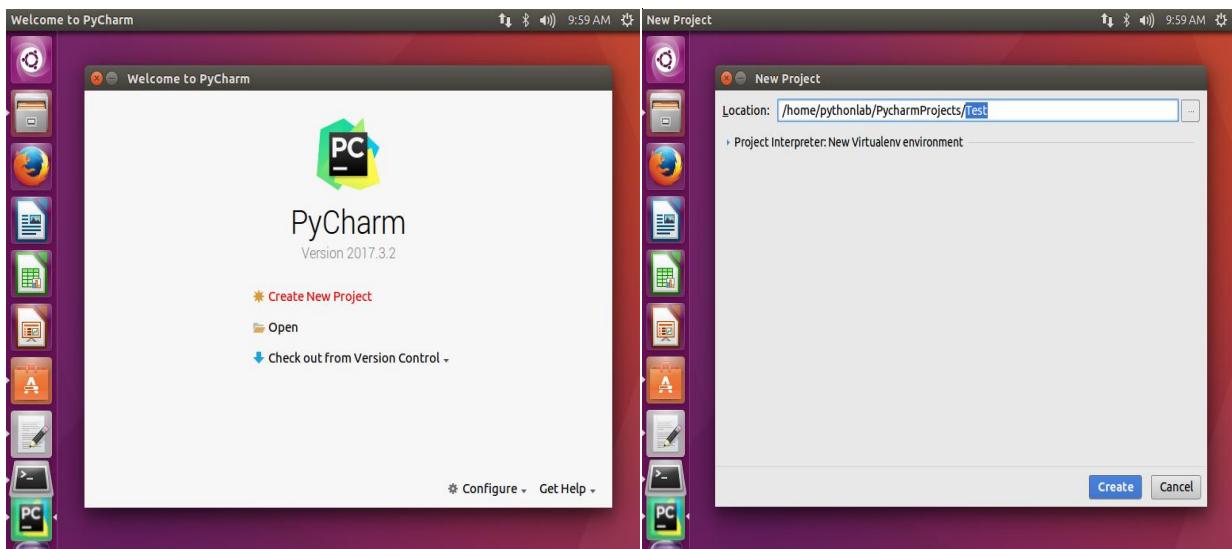
STEPS TO INSTALL PYCHARM

Step 1:- Download and Install Pycharm from <https://www.jetbrains.com/pycharm/download> during download please select community edition (installation is same as python in Ubuntu)

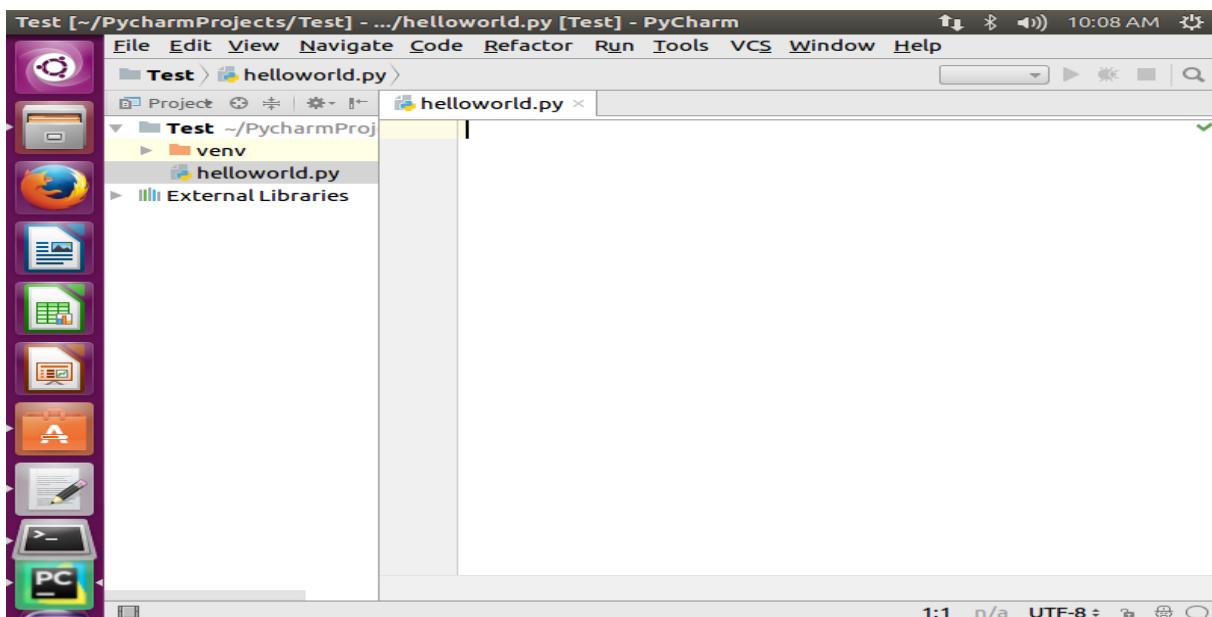
```
Terminal pythonlab@ubuntu: ~/Downloads
pythonlab@ubuntu:~$ cd Downloads/
pythonlab@ubuntu:~/Downloads$ ls
pycharm-community-2017.3.2.tar.gz  Python-3.6.4.tgz
Python-3.6.4
pythonlab@ubuntu:~/Downloads$ tar-xzvf pycharm-community-2017.3.2
.tar.gz
```

A screenshot of a terminal window titled "Terminal". The window shows a file listing in the "/Downloads/pycharm-community-2017.3.2/bin" directory. The files listed are: format.sh, idea.properties, pycharm64.vmoptions, restart.py, fsnotifier, inspect.sh, pycharm.png, fsnotifier64, log.xml, pycharm.sh, fsnotifier-arm, printenv.py, and pycharm.vmoptions. Below the listing, the command "./pycharm.sh" is typed and highlighted with a cursor. In the background, a large window for PyCharm Community Edition 2017.3 is visible, displaying its splash screen with the PyCharm logo and version information.

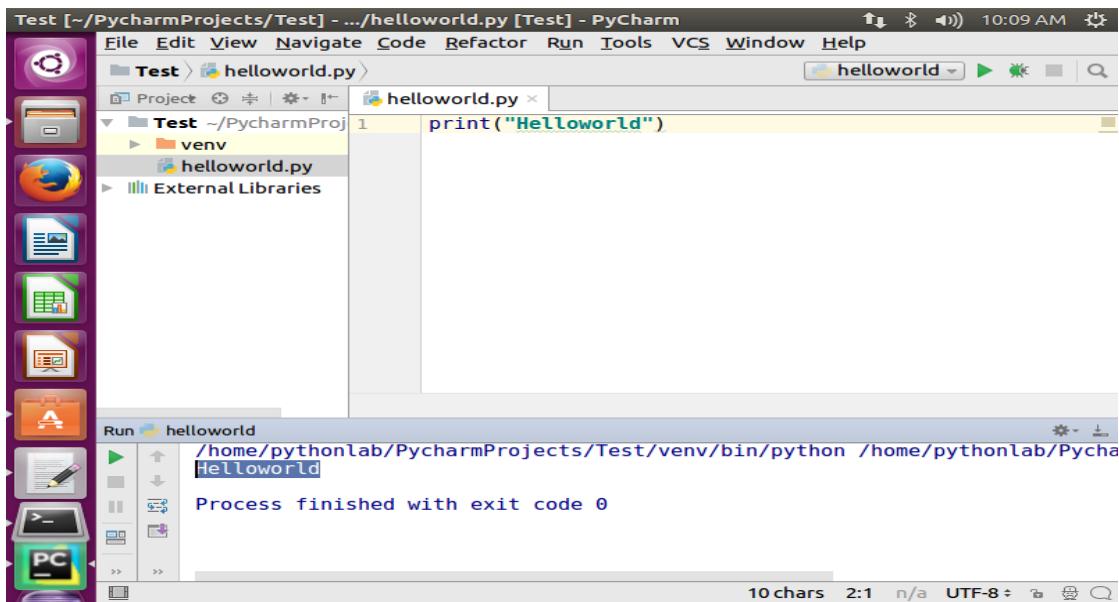
Step 2:- Once Pycharm is ready then Click on Create new project and Specify a project name like test in this example and click on Create.



Step 3:- Rt click on Test project and select New Python file, specify its name to it like helloworld.py in this example.



Step 4:- Write and test application in Pycharm



```
Test [~/PycharmProjects/Test] - .../helloworld.py [Test] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Test > helloworld.py
Project Test ~/PycharmProj venv helloworld.py External Libraries
1 print("HelloWorld")
Run helloworld
/home/pythonlab/PycharmProjects/Test/venv/bin/python /home/pythonlab/PycharmProjects/Test/helloworld.py
HelloWorld
Process finished with exit code 0
```

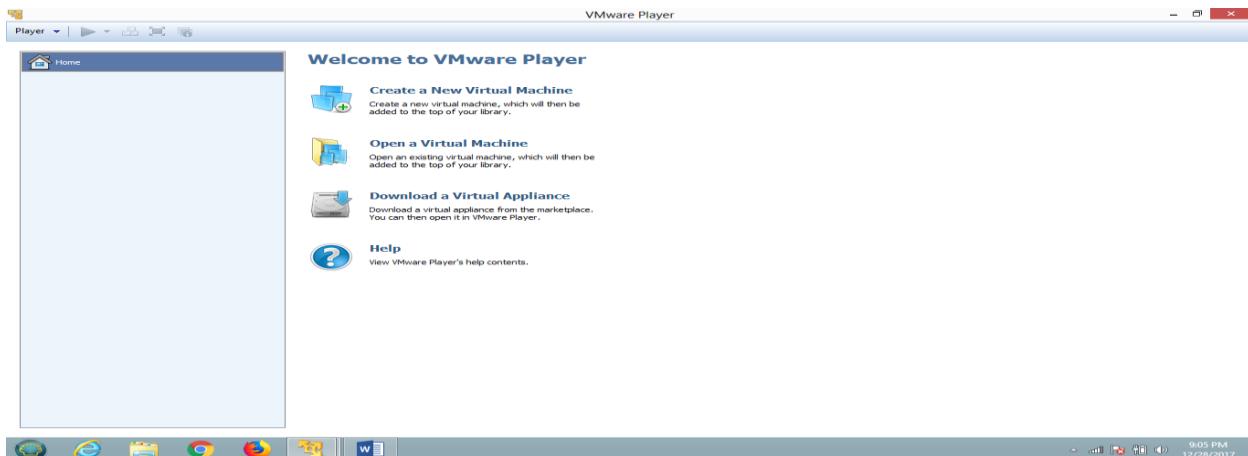
STEPS TO OPEN PYTHON VIRTUAL MACHINE

Step 1 :- Download VMware Player from following Link or search it on Google for alternate links.

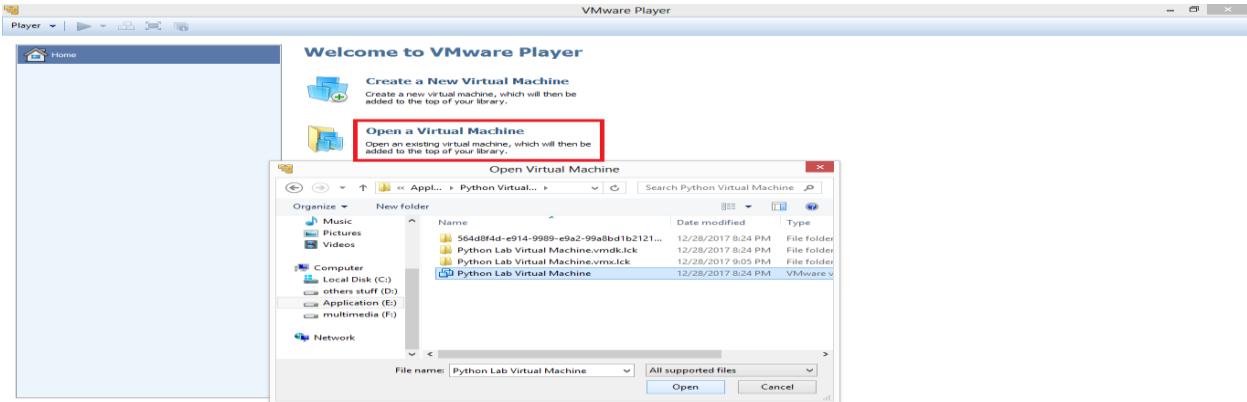
https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0

Once package is downloaded then install it.

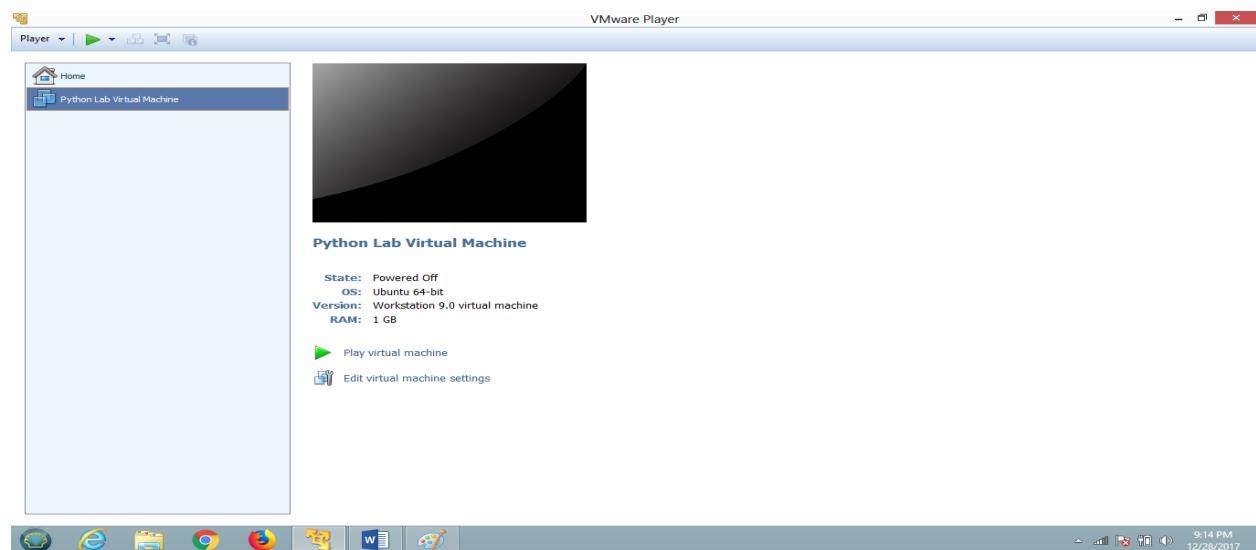
Step 2 :- Once Installation is done then Open it.



Step 3 :- Click on Open Virtual machine and Locate VMDK file (Copy Python Virtual machine folder in C drive and locate VMDK file in Given Folder)



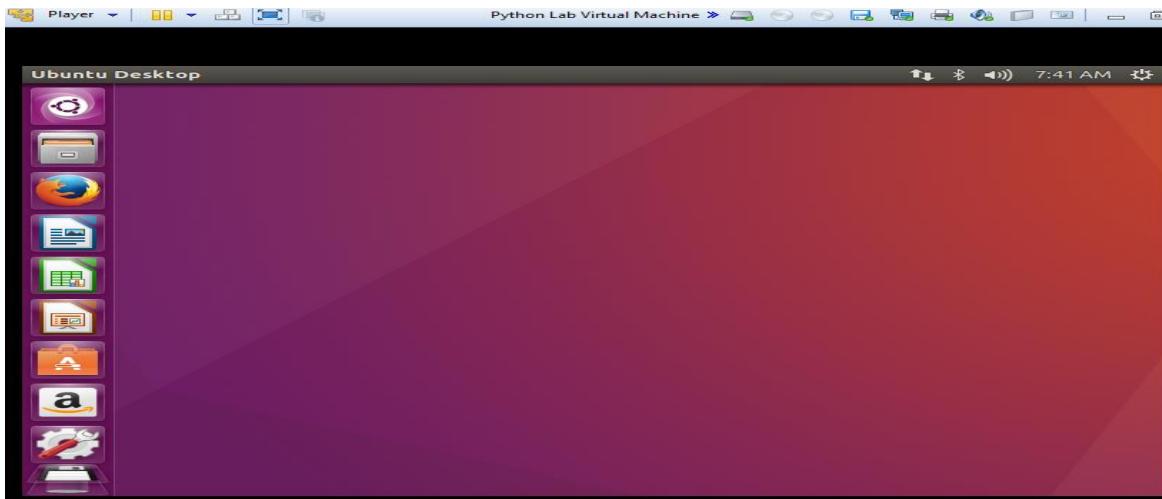
Step 4 :- Select Python Lab Virtual Machine and Click on play Virtual machine.



Step 5 :- Specify the credentials as below

Username :- pythonlab

Password :- python



Implemented Programs

EXPERIMENT NO 1

1.1 Write python programs to understand Comments, Datatypes, Expressions, Input and Output Functions

```
print "-----COMMENTS-----"
print 'Single Line #'"
print 'Multi Line ("---")'
print "Multi Line ('---')"
print """
print "--Datatypes,Expressions,Output Statement & Conversions-"
a=16 #int datatype
a1="abc" #string datatype
b=23.32 #Float datatype
c=2.3+3.2j #Complex datatype
d=1.2-2.8j #Complex datatype
e=c+d #expression
e1=a*b #expression
print "Sum is",e #Output statement
print 'Multiplication is ', e1 #Output statement
f=float(a) #Converting int to float
print 'Int to Float', f
g=complex(b) #Converting float to complex
print 'Float to Complex', g
h=oct(a) #Decimal to octal
```

```

print 'Decimal to Octal', h
i=bin(a) #Decimal to binary
print 'Decimal to Binary', i
j=hex(a) #Decimal to hexadecimal
print 'Decimal to Hexadecimal',j
print ""
print "-----INPUT STATEMENT-----"
k=raw_input('What is your name? using raw input ') #ask & returns string of data
print 'Your name is ' + k
l=input('Enter value of x ')
print 'Entered value of x using String Function is ' + str(l) #Printing Output in String Format
print 'Entered value of x using format function is {0}'.format(l) #Printing Formatted Output
print "accept integer input",int(input()) #Accept Integer Input

print ""
print "-----Knowing the datatype-----"
a=12
print type(a)
b="str"
print type(b)
c=1.2+4j
print type(c)

```

OUTPUT OF PROGRAM 1.1

```

pythonlab@ubuntu:~/My experiments/Experiment 1$ python 1.1.py
-----COMMENTS-----
Single Line "#"
Multi Line ("""---""")
Multi Line ('''----''')

--Datatypes, Expressions, Output Statement & Conversions-
Sum is (3.5+0.4j)
Multiplication is 373.12
Int to Float 16.0
Float to Complex (23.32+0j)
Decimal to Octal 020
Decimal to Binary 0b10000
Decimal to Hexadecimal 0x10

-----INPUT STATEMENT-----
What is your name? using raw input BHUSHAN JADHAV
Your name is BHUSHAN JADHAV
Enter value of x 23
Entered value of x using String Function is 23
Entered value of x using format function is 23
accept integer input 21
21

-----Knowing the datatype-----
<type 'int'>
<type 'str'>
<type 'complex'>

```

1.2 Write python programs to understand Byte array, Range, Set and STRING Functions

```
#Demonstrating
print "-----BYTEARRAY FUNCTIONS-----"
print ""
x=[10,20,30,40,50] #Creating integer List
print "Creating a bytearray using bytearray function"

x1=bytearray(x) #Converting List in to Bytearray
print "printing bytearray:"
print(x[0],x[1],x[2],x[3],x[4]) #printing Bytearray
print ""

print "-----RANGE DATATYPE -----"
print ""
r=range(1,9) #Creating a range of 1 to 9 elements
print "range of 1 to 9 is :",r
r=range(10) #Creating a range first 10 elements
print "range of 10 is :",r
print ""

print "--SET DATATYPE (Unordered elements with no duplicate)--"
print ""
s={1,4,6,3} #Creating a set
print s #Printing set in ascending order
c1=set("BHUSHANA") #Creating a character set
print c1 #Printing set alphabetically and avoiding duplication of Character
s.update([4,5]) #Adding element 5 after 4 in set s
print s #Printing set in ascending order
fs=frozenset(s) #Creating a frozen of set s i.e. Fixed Set cant be modified
print fs
print ""

print "-----STRING -----"
print ""
str="""WELCOME TO 'CORE PYTHON PROGRAMMING' COURSE"""\ #STRING Datatype
print "printing String array:"
print str
print "Printing first four element of String Array:" #Printing first four element of String Array
#FIRST POSITION WILL BE A SPACE SO PRINT ACCORDINGLY
print str[0:4]
print "Printing first character from end:" #Printing first character from end
print str[-1]
```

```

print "Printing nineth character of String Array:" #Printing nineth
character
print str[8]
print ""

print "-----STRING FUNCTIONS-----"
print ""

s1="CORE PYTHON PROGRAMMING"
s2="IS BEST"
print "Length:", len(s1) #Printing length
print "Repeate twice:",(s1)*2 #Repeat string
print "Concat : specified Character in the strings:",s1.count('M')
#Counting
a1='CORE'
a2='BEST'
print "Replacing CORE with BEST:", s1.replace(a1,a2)
print "Changing case from uppercase to Lower:", s1.lower()
print "SORTING STRINGS:", sorted(s2)
s4=u'\u0915\u0947\u0964\u0930 \u092a\u0948\u0925\u0964\u0928'
#String with unicode
print "UNICODE FOR CORE PYTHON", s4

```

OUTPUT OF PROGRAM 1.2

```

pythonlab@ubuntu:~/My experiments/Experiment 1$ python 1.2.py
-----BYTEARRAY FUNCTIONS-----
Creating a bytearray using bytearray function
printing bytearray:
(10, 20, 30, 40, 50)

-----RANGE DATATYPE -----
range of 1 to 9 is : [1, 2, 3, 4, 5, 6, 7, 8]
range of 10 is : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

--SET DATATYPE (Unordered elements with no duplicate)--
set([1, 3, 4, 6])
set(['A', 'B', 'H', 'N', 'S', 'U'])
set([1, 3, 4, 5, 6])
frozenset([1, 3, 4, 5, 6])

-----STRING -----
printing String array:
WELCOME TO 'CORE PYTHON PROGRAMMING' COURSE
Printing first four element of String Array:
WELC
Printing first character from end: E
Printing nineth character of String Array: T

-----STRING FUNCTIONS-----
Length: 23
Repeate twice: CORE PYTHON PROGRAMMINGCORE PYTHON PROGRAMMING
Concat : specified Character in the strings: 2
Replacing CORE with BEST: BEST PYTHON PROGRAMMING
Changing case from uppercase to Lower: core python programming
SORTING STRINGS: [' ', 'B', 'E', 'I', 'S', 'S', 'T']
UNICODE FOR CORE PYTHON કોર્એ પેથાન

```

1.3 Write python programs to understand List Touple, Dictionaries and Arrays

```
from array import * #Import an array package
print "-----LIST -----"
print ""
l=[12,23,-5,0.8,'python','BJ'] #Creating a list of variable datatypes
print "printing original list",l #printing list
print "printing first three elements of a list", l[0:3] #printing first three elements of a list
print "printing Last elements of a list",l[-1] #printing Last elements of a list
print "printing first three elements of a list twice",l[0:3] * 2 #printing first three elements of a
list twice
print ""

print "-----LIST FUNCTIONS-----"
print ""
l1=list(range(1,8)) #Creating a list of range 1 to 7
print "List of range 1 to 7",l1
l1.append(9) #append 9
print "Append 9 :",l1

l1.sort(reverse=True)
print "Descending Sort:",l1
l1.sort()
print "Ascending Sort:",l1
l1.reverse()
print "Reverse:",l1
l1.sort()
l1.remove(9)
print "Remove 9:",l1
print "count:",l1.count(5)
print "max:",max(l1)
print "min:",min(l1)
print "Index of 6:",l1.index(6)

print "-----MATRICES using LIST-----"
print ""
m1=[[1,2,3],[4,5,6],[7,8,9]]
for r in m1:
    print(r)

print ""
print "-----TOUPLE DATATYPE (READ ONLY LIST)-----"
print ""
tpl=(-5,0.8,'python','BJ') #Creating a Touple
print "Created Touple is ", tpl
print "Touple elements 0 to 2 is", tpl[0:2] #printing first two elements of a tuple
```

```

print "Touple element -2 is ", tpl[-2] #printing Second last elements of a tuple
print ""
print "-----TOUPLE FUNCTIONS -----"
tpl2=(10,20,30,40,10,20,10) #Creating a Tuple2
print "Created Tuple2 is ", tpl2
print "Length : ", len(tpl2)
print "Min : ", min(tpl2)
print "Max : ", max(tpl2)
print "Count no. of 10's: ", tpl2.count(10)
print "Reverse Sort : ", sorted(tpl2,reverse=True)

print ""
print "-----DICTIONARIES i.e. key:value pair-----"
print ""
d1={'Name':'bhushan','gender':'M','age':32,'college':'tsec'} #create dictionary
print "print dictionary: ",d1
print ""
print "-->Keys :",d1.keys()
print "-->Values :",d1.values()
print "-->Keys and values :",d1.items()
d1.update({'country':'india'})
print "-->Print updated dictionary: ",d1
c=sorted(d1.items(),key=lambda t:t[1])
print "-->Sort By values :",c

print "-----ARRAYS -----"
print ""

arr=array('i',[10,20,30,40,50]) #Create array with integer values
print "The Array Elements are"
for i in arr: #i is element and arr in array name
    print i #Requires indentation
print "length of array is",len(arr)

arr1=array('c',['a','b','c','d']) #Create array with character values
print "The Character Array Elements are"
for ch in arr1: #i is element and arr1 in array name
    print ch #Requires indentation

```

OUTPUT OF PROGRAM 1.3

-----LIST -----

```

printing original list [12, 23, -5, 0.8, 'python', 'BJ']
printing first three elements of a list [12, 23, -5]

```

```

printing Last elements of a list BJ
printing first three elements of a list twice [12, 23, -5, 12, 23, -5]

-----LIST FUNCTIONS-----

List of range 1 to 7 [1, 2, 3, 4, 5, 6, 7]
Append 9 : [1, 2, 3, 4, 5, 6, 7, 9]
Descending Sort: [9, 7, 6, 5, 4, 3, 2, 1]
Ascending Sort: [1, 2, 3, 4, 5, 6, 7, 9]
Reverse: [9, 7, 6, 5, 4, 3, 2, 1]
Remove 9: [1, 2, 3, 4, 5, 6, 7]
count: 1
max: 7
min: 1
Index of 6: 5
-----MATRICES using LIST-----

[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

-----TOUPLE DATATYPE (READ ONLY LIST)-----

Created Touple is (-5, 0.8, 'python', 'BJ')
Touple elements 0 to 2 is (-5, 0.8)
Touple element -2 is python

-----TOUPLE FUNCTIONS -----
Created Touple2 is (10, 20, 30, 40, 10, 20, 10)
Length : 7
Min : 10
Max : 40
Count no. of 10's: 3
Reverse Sort : [40, 30, 20, 20, 10, 10, 10]

-----DICTIONARIES i.e. key:value pair-----

print dictionary: {'gender': 'M', 'age': 32, 'college': 'tsec',
'Name': 'bhushan'}

-->Keys : ['gender', 'age', 'college', 'Name']
-->Values : ['M', 32, 'tsec', 'bhushan']
-->Keys and values : [('gender', 'M'), ('age', 32), ('college',
'tsec'), ('Name', 'bhushan')]
-->Print updated dictionary: {'gender': 'M', 'age': 32, 'college':
'tsec', 'Name': 'bhushan', 'country': 'india'}
-->Sort By values : [('age', 32), ('gender', 'M'), ('Name',
'bhushan'), ('country', 'india'), ('college', 'tsec')]
-----ARRAYS -----

```

```

The Array Elements are
10
20
30
40
50
length of array is 5
The Character Array Elements are
a
b
c
d

```

Experiment 2:

2.1 Write python programs to understand the Control Structures

```

#Control Structures remember the indentations
print "--Finding MAX BETWEEN 3 Nos using IF-Else-Elif STATEMENT--"
a=input("Enter value of First element:")
b=input("Enter value of Second element:")
c=input("Enter value of Third element:")
if a>b and a>c:
    print a,"is Greater"
elif b>c:
    print b,"is Greater"
else:
    print c,"is Greater"
print ""

print "-----Calculate Factorial using WHILE LOOP-----"
num=int(input("Enter the number : "))
fact=1
i=1
while i <= num:
    fact=fact*i
    i=i+1
print"Factorial of",num," is :",fact

print "-----Generate Fibonacci series using If statement & WHILE LOOP--"
n=int(input("How many numbers??? : "))
f1=0
f2=1
c=2
if n==1:
    print f1
elif n==2:
    print f1,'\n',f2
else:
    print f1,'\n',f2

```

```

while c<n:
    f=f1+f2
    print f
    f1,f2=f2,f
    c+=1

-----OUTPUT-----

pythonlab@ubuntu:~/PYTHON ALL EXPERIMENTS$ python 2.1.py
--Finding MAX BETWEEN 3 Nos using IF-Else-Elif STATEMENT--
Enter value of First element:43
Enter value of Second element:43
Enter value of Third element:2
43 is Greater

-----Calculate Factorial using WHILE LOOP-----
Enter the number : 2
Factorial of 2 is : 2
-----Generate Fibonacci series using If statement & WHILE LOOP-----
--
How many numbers??? : 10
0
1
1
2
3
5
8
13
21
34

```

2.2 Write python programs to understand the For Loop

```

#FOR LOOP
print "--Multiplication table using FOR loop--"
n=int(input("Enter the number:"))
for i in range(1,11):
    print n,"X",i,"=",n*i
print ""
print "--Pascal triangles--"

n=int(input("Enter the rows:"))
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)]
    n>=1

```

-----**OUTPUT**-----

```
pythonlab@ubuntu:~/PYTHON ALL EXPERIMENTS$ python 2.2.py
```

```
--Multiplication table using FOR loop--
```

```
Enter the number:3
```

```
3 X 1 = 3
```

```
3 X 2 = 6
```

```
3 X 3 = 9
```

```
3 X 4 = 12
```

```
3 X 5 = 15
```

```
3 X 6 = 18
```

```
3 X 7 = 21
```

```
3 X 8 = 24
```

```
3 X 9 = 27
```

```
3 X 10 = 30
```

```
--Pascal triangles--
```

```
Enter the rows:6
```

```
[1]
```

```
[1, 1]
```

```
[1, 2, 1]
```

```
[1, 3, 3, 1]
```

```
[1, 4, 6, 4, 1]
```

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

2.3 Write python programs to understand Functions

```
#Functions to run basic calculator
def menu(): #Defining function Menu
    #print what options you have
    print ""
    print "Welcome to calculator in Python"
    print "your options are:"
    print " "
    print "1) Addition"
    print "2) Subtraction"
    print "3) Multiplication"
    print "4) Division"
    print "5) Quit calculator"
    print " "
    return input ("Choose your option: ")

# this adds two numbers given
def add(a,b): #Defining function add
    print a, "+", b, "=", a + b
```

```

# this subtracts two numbers given
def sub(a,b): #Defining function sub
    print a, "-", b, "=", a - b

# this multiplies two numbers given
def mul(a,b): #Defining function mul
    print a, "*", b, "=", a * b

# this divides two numbers given
def div(a,b): #Defining function div
    print a, "/", b, "=", a / b

# NOW THE PROGRAM STARTS, AS CODE IS RUN
loop = 1
choice = 0

while loop == 1:
    choice = menu()
    if choice == 1:
        num1=int(input("Enter num 1 --: "))
        num2=int(input("Enter num 2 --: "))
        add(num1,num2)
    elif choice == 2:
        num1=int(input("Enter num 1 --: "))
        num2=int(input("Enter num 2 --: "))
        sub(num1,num2)
    elif choice == 3:
        num1=int(input("Enter num 1 --: "))
        num2=int(input("Enter num 2 --: "))
        mul(num1,num2)
    elif choice == 4:
        num1=int(input("Enter num 1 --: "))
        num2=int(input("Enter num 2 --: "))
        div(num1,num2)
    elif choice == 5:
        loop = 0

# End of the program

```

OUTPUT OF PROGRAM 2.3

pythonlab@ubuntu:~/PYTHON ALL EXPERIMENTS\$ python 2.3.py

Welcome to calculator in Python

your options are:

- 1) Addition
- 2) Subtraction
- 3) Multiplication
- 4) Division
- 5) Quit calculator

Choose your option: 1

Enter num 1 --: 21

Enter num 2 --: 2

$21 + 2 = 23$

Welcome to calculator in Python

your options are:

- 1) Addition
- 2) Subtraction
- 3) Multiplication
- 4) Division
- 5) Quit calculator

Choose your option: 2

Enter num 1 --: 23

Enter num 2 --: 2

$23 - 2 = 21$

Welcome to calculator in Python

your options are:

- 1) Addition
- 2) Subtraction
- 3) Multiplication
- 4) Division
- 5) Quit calculator

Choose your option: 3

Enter num 1 --: 23

Enter num 2 --: 1

$23 * 1 = 23$

Welcome to calculator in Python

your options are:

- 1) Addition
- 2) Subtraction

- 3) Multiplication
- 4) Division
- 5) Quit calculator

Choose your option: 23

Welcome to calculator in Python
your options are:

- 1) Addition
- 2) Subtraction
- 3) Multiplication
- 4) Division
- 5) Quit calculator

Choose your option: 4

Enter num 1 --: 23

Enter num 2 --: 3

23 / 3 = 7

Welcome to calculator in Python
your options are:

- 1) Addition
- 2) Subtraction
- 3) Multiplication
- 4) Division
- 5) Quit calculator

Choose your option: 5

Experiment No: 3

3.1 Write python programs to understand Classes, object, Static method and inner class

```
#Demonstrate Class object static method and inner class
class Student:
    roll=int(input('Enter your Roll Number \n'))
    name=raw_input('Enter your Name \n')
    age=int(input('Enter your Age \n'))
    gender=raw_input('Enter your Gender \n')
    print ""
    class address: #Inner Class
        print "-----Inner Class-----"
        print "This is Inner class"
```

```

@staticmethod #Static method
def statmeth():
    print "This is Static method"

def display_record(self):
    print "Your Roll number is",self.roll
    print "Your Name is",self.name
    print "Your Age is",self.age
    print "Your Gender is",self.gender
    print ""

```

```

s=Student() #Main class object
print ""
print "-----Static Method-----"
s.statmeth() #calling static method
print ""
print "-----Details of Student are-----"
s.display_record() #Calling class method

```

-----OUTPUT-----

```

pythonlab@ubuntu:~$ python 3.1.py
Enter your Roll Number
1
Enter your Name
BHUSHAN JADHAV
Enter your Age
32
Enter your Gender
MALE

```

-----Inner Class-----

This is Inner class

-----Static Method-----

This is Static method

-----Details of Student are-----

```

Your Roll number is 1
Your Name is BHUSHAN JADHAV
Your Age is 32
Your Gender is MALE

```

3.2 Write python programs to understand Constructors

```

#Demonstrating Constructor

class Student:
    def __init__(self,n="",m1=0,m2=0,m3=0,m4=0,m5=0): #Constructor
        self.name=raw_input('Enter your name \n')
        self.hmarks=int(input('Enter your History marks \n'))
        self.mmarks=int(input('Enter your Maths marks \n'))
        self.smarks=int(input('Enter your Science marks \n'))
        self.gmarks=int(input('Enter your Geography marks \n'))
        self.amarks=float(self.hmarks+self.mmarks+self.smarks
        +self.gmarks)/4

    def display_record(self): #Instance method
        print "Hello Mr.",self.name
        print "Your Average marks are",self.amarks
        if(self.amarks>=75):
            print "You got Distinction class"

        elif(self.amarks>=60):
            print "You got First class"

        elif(self.amarks>=50):
            print "You got Second class"

        elif(self.amarks>=40):
            print "You got Pass class"

        else:
            print "You are failed"
    s=Student()
    print ""
    print "-----RESULT-----"
    s.display_record()

```

-----OUTPUT-----

```

pythonlab@ubuntu:~$ python 3.2.py
Enter your name
BHUSHAN JADHAV
Enter your History marks
87
Enter your Maths marks
83
Enter your Science marks
85
Enter your Geography marks

```

-----RESULT-----

Hello Mr. BHUSHAN JADHAV
 Your Average marks are 83.25
 You got Distinction class

3.3 Write python programs to understand Inheritance and Polymorphism with Method overloading and Method Overriding

```
class Student:      # define parent class
    def __init__(self):
        print "Calling parent constructor"

    def teacher(self): #Overridden Method (Method Over riding)
        print "Overriding Teacher method in Parent Class"

    def s_details(self):
        print ".....CALLING PARENT METHOD using Child Object....."
        self.roll=raw_input("Enter Your Roll Number. :- ")
        self.name=raw_input("Enter Your Name :- ")
        self.div=raw_input("Enter Your Division :- ")
        self.city=raw_input("Enter Your City :- ")
        print ""
        print "Student Details Are...."
        print "Your Roll Number is",self.roll
        print "Your Name is",self.name
        print "Your Division is",self.div
        print "Your City is",self.city

    def sum(self, a=None,b=None,c=None):
        print ""

        print ".....METHOD OVERLOADING WITH a=10,b=20 and c=30....."

    if a is not None and b is not None and c is not None :
        print 'Sum of a,b and c is',a+b+c
    elif a is not None and b is not None:
        print 'Sum of a and b is',a+b
    else:
        print 'Sum Not possible..Value of a is',a

class College(Student): # define child class (Inheritance)
    def __init__(self):
        print "Calling child constructor"
```

```

def teacher(self): #Overridden Method (Method Over riding)
    print ""
    print ".....METHOD OVERRIDING:Overriding Teacher method in Child Class"

def c_details(self):
    print ".....CALLING CHILD METHOD using Child Object....."
    self.cname=raw_input("Enter Your College Name :- ")
    self.uni=raw_input("Enter Your University Name :- ")
    self.caddress=raw_input("Enter Your College City :- ")
    print ""
    print "Your College Details Are....."
    print "Your College name is",self.cname
    print "Your University is",self.uni
    print "Your College City is",self.caddress

c = College()      # instance of child
c.c_details()     # calls parent's method
c.s_details()     # calls parent's method
c.teacher()       # calls Overridden method
c.sum(10,20,30)   # Method overloading with three parameters
c.sum(10,20)      # Method overloading with two parameters
c.sum(10)         # Method overloading with one parameters

```

-----OUTPUT-----

```

pythonlab@ubuntu:~$ python 3.3.py
Calling child constructor
.....CALLING CHILD METHOD using Child Object.....  

Enter Your College Name :- TSEC  

Enter Your University Name :- MUMBAI  

Enter Your College City :- BANDRA

```

```

Your College Details Are.....  

Your College name is TSEC  

Your University is MUMBAI  

Your College City is BANDRA  

.....CALLING PARENT METHOD using Child Object.....  

Enter Your Roll Number. :- 1  

Enter Your Name :- BHUSHAN JADHAV  

Enter Your Division :- A  

Enter Your City :- THANE

```

```

Student Details Are.....  

Your Roll Number is 1  

Your Name is BHUSHAN JADHAV

```

Your Division is A
Your City is THANE

.....METHOD OVERRIDING:Overriding Teacher method in Child Class

.....METHOD OVERLOADING WITH a=10,b=20 and c=30.....
Sum of a,b and c is 60

.....METHOD OVERLOADING WITH a=10,b=20 and c=30.....
Sum of a and b is 30

.....METHOD OVERLOADING WITH a=10,b=20 and c=30.....
Sum Not possible..Value of a is 10

3.4 Write python program to understand different types of Exceptions

```
i=1
while i<=5:
    n=int(input("Please enter numbers between 1 to 5 to see diffrent Exceptions : "))
    if n==1:
        try:
            a=int(input("Please enter number a : "))
            b=int(input("Please enter number b (put b=0): "))
            c=a/b
        except ZeroDivisionError:
            print("Oops! Number Divisible by Zero Exception Occurs.")
        else:
            print "Division is",c

    elif n==2:
        try:
            a=int(input("Please enter number a : "))
            b=int(input("Please enter number b (put b='a'): "))
            c=a/b
        except ValueError:
            print("Oops! Value Error Exception Occurs.Please enter a valid number.")
        else:
            print "Division is",c

    elif n==3:
        try:
            a=int(input("Please enter number a : "))
            b=int(input("Please enter number b : "))
            c=k/b
        except NameError:
```

```

print("Oops! Name Error Exception Occurs due to c=k/b (k is not defined ).Please enter a
valid variable number.")

elif n==4:
    try:
        r='2'+2
    except TypeError:
        print("Oops! Type Error Exception Occurs (due to '2'+2).Please Provide Valid data type. ")

elif n==5:
    try:
        n=int(input("Please enter Numbers between 2 to 3: (Check for other nos) "))
        assert n>=2 and n<=3
        print("The Number Entered is",n)
    except AssertionError:
        print("Oops! Assertion Error Occurs..Please enter number between 2 to 5.")

else:
    print "Existing The Program"
    exit()
i+=1

```

-----OUTPUT-----

```

pythonlab@ubuntu:~$ python 3.4.py
Please enter numbers between 1 to 5 to see diffrent Exceptions : 1
Please enter number a : 12
Please enter number b (put b=0): 0
Oops! Number Divisible by Zero Exception Occurs.
Please enter numbers between 1 to 5 to see different Exceptions : 2
Please enter number a : 24
Please enter number b (put b='a'): 'a'
Oops! Value Error Exception Occurs.Please enter a valid number.
Please enter numbers between 1 to 5 to see different Exceptions : 3
Please enter number a : 13
Please enter number b : 45
Oops! Name Error Exception Occurs due to c=k/b (k is not defined ).Please enter a valid
variable number.
Please enter numbers between 1 to 5 to see different Exceptions : 4
Oops! Type Error Exception Occurs (due to '2'+2).Please Provide Valid data type.
Please enter numbers between 1 to 5 to see different Exceptions : 5
Please enter Numbers between 2 to 3: (Check for other nos) 6
Oops! Assertion Error Occurs. Please enter number between 2 to 5.
Please enter numbers between 1 to 5 to see different Exceptions : 6
Existing The Program

```

Experiment No: 4

4.1 Write python programs to understand different file handling operations with pickle

Filename:- studentpickle.py

```
class student:  
    def __init__(self,roll,name,age):  
        self.roll=roll  
        self.name=name  
        self.age=age  
    def display(self):  
        print('{:5d}{:20s}{:10.2f}'.format(self.roll,self.name,self.age))
```

Filename:- 4.1.py

```
import pickle,studentpickle  
  
print "*****WRITE file*****"  
f=open('myfile.txt','w') #Create file if not exist and open it for writing  
s1=raw_input('Enter Your Text to be written in to file: ')  
f.write(s1) #write text into file  
f.close() #Close file  
print"  
print "*****Read file *****"  
f1=open('myfile.txt','r') #open file for reading  
print "Contents of file are"  
s=f1.read()  
print s  
f1.close()#Close file  
print"  
print "*****Append file*****"  
f=open('myfile.txt','a') #open file for Appending  
s2=raw_input('Enter text to be appended in original file..')  
f.write(s2)  
f.close() #Close file  
print"  
print ""  
print "*****Read Appended file*****"  
f1=open('myfile.txt','r') #Create file if not exist and open it for reading  
s=f1.read()  
print s  
f.close()  
  
print "*****WITH OPEN*****"
```

```

with open('sample.txt','w') as f:
    s3=raw_input('Enter text for WITHOPEN write..')
    f.write(s3)
with open('sample.txt','r') as f:
    for line in f:
        print line

print "*****PICKLE*****"
f=open('studentpickle.dat','wb')
n=int(input('How many students?? '))
for i in range (n):
    roll=int(input("Enter your roll no : "))
    name=raw_input("Enter your name : ")
    age=int(input("Enter your age : "))
    s=studentpickle.student(roll,name,age)
    pickle.dump(s,f)
f.close()

```

OUTPUT

pythonlab@ubuntu:~/PYTHON ALL EXPERIMENTS/Experiment 4\$ python 4.1.py
*****WRITE file*****

Enter Your Text to be written in to file: BHUSHAN

*****Read file *****

Contents of file are

BHUSHAN

*****Append file*****

Enter text to be appended in original file.. JADHAV

*****Read Appended file*****

BHUSHAN JADHAV

*****WITH OPEN*****

Enter text for WITHOPEN write.. PYTHON

PYTHON

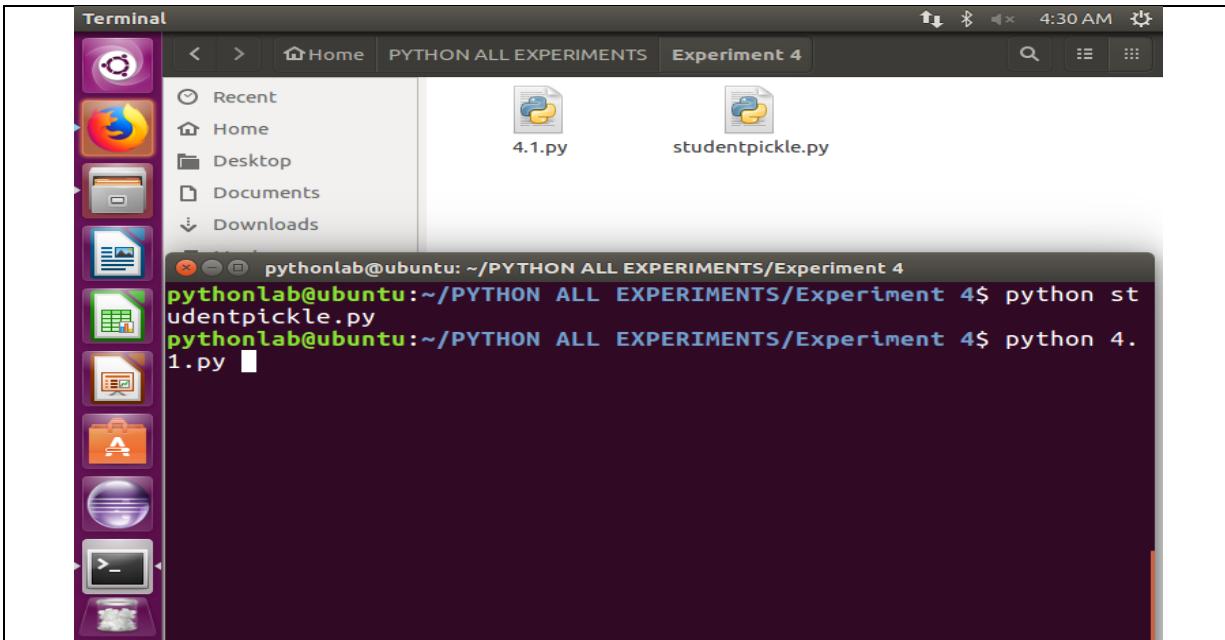
*****PICKLE*****

How many students?? 1

Enter your roll no : 1

Enter your name : BHUSHAN

Enter your age : 32



4.2 Write python programs to understand Lambda, map, reduce, filter and range functions

```
"-----Lambda function-----"
print "Lambda Function"
f = lambda x, y : x + y
print f(1,1)
print ""

"-----Map function-----"
print "Map Function"
def fahrenheit(T):
    return ((float(9)/5)*T + 32)
def celsius(T):
    return (float(5)/9)*(T-32)
temp = (36.5, 37, 37.5, 39)
F = map(fahrenheit, temp)
C = map(celsius, F)
print F
print C
print ""

"-----Reduce function-----"
print "Reduce Function"
print reduce(lambda x,y: x+y, [1,2,3,4])
print ""

"-----Filter function-----"
```

```

print "Filter Function"
fib = [0,1,1,2,3,5,8,13,21,34,55]
result = filter(lambda x: x % 2, fib)
print result
print ""

"-----Range function-----"
print "Range Function"
print range(10)

```

OUTPUT

```

pythonlab@ubuntu:~$ python 4.2.py
Lambda Function
2

Map Function
[97.7, 98.60000000000001, 99.5, 102.2]
[36.5, 37.00000000000001, 37.5, 39.0]

```

Reduce Function
10

Filter Function
[1, 1, 3, 5, 13, 21, 55]

Range Function
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Experiment No: 5

5.1 Write python programs to understand GUI Canvas Application using Tkinter

```

from Tkinter import *

root=Tk()
root.title("My First Tkinter Application") #Giving a title to bar
c=Canvas(root,bg="orange",height=700, width=800) #Creating a canvas
id=c.create_line(80,80,200,80,200,200,width=6,fill="cyan3")#Drawing a Line
id=c.create_line(80,80,300,80,300,300,width=6,fill="cyan3")#Drawing a Line
id=c.create_oval(120,120,400,300,width=6,fill="red",outline="cornsilk1",activefill="green")# Drawing a Oval
fnt=('Times',22,'bold italic underline') #Setting Font

```

```

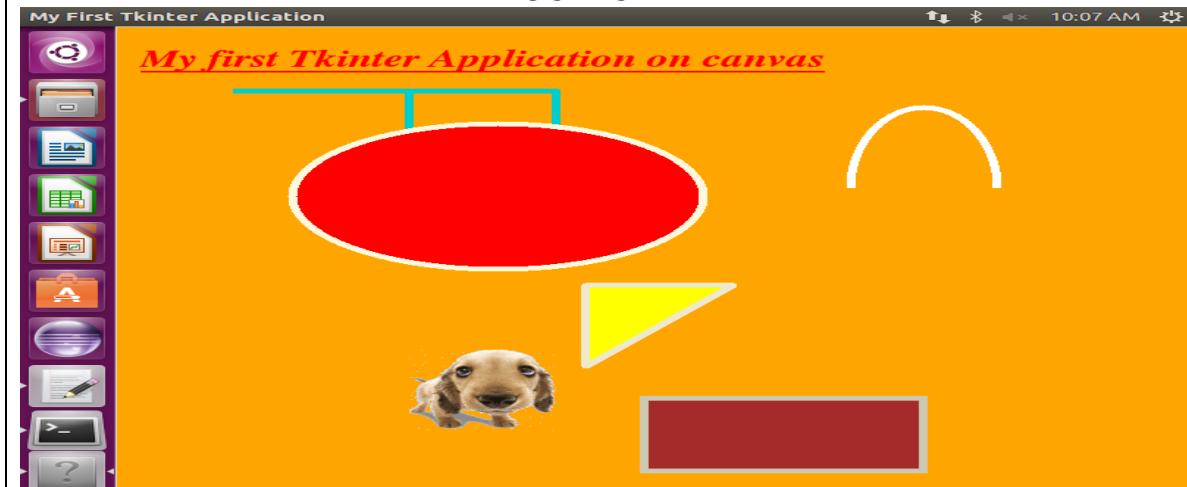
id=c.create_text(250,40,text="My first Tkinter Application on
canvas",font=fnt,fill="red",activefill="green") #Drawing a Text
id=c.create_polygon(320,320,320,420,420,320,width=6,fill="yellow",outline="cornsilk2",acti
vefill="lightblue")#Drawing Polygon

id=c.create_rectangle(360,460,550,550,width=6,fill="brown",outline="cornsilk3",activefill="c
hocolate")#Drawing Rectangle
id=c.create_arc(500,100,600,300,width=6,start=0,
extent=180,outline="white",style="arc")#Drawing arc
file1=PhotoImage(file="cat.gif",height=100,width=100)
file2=PhotoImage(file="dog.gif",height=100,width=100)
id=c.create_image(300,400,anchor=NE,image=file2,activeimage=file1)#Drawing Image

c.pack()
root.mainloop()

```

-----OUTPUT-----



5.2 Write python programs to understand GUI Frame Application using Tkinter

5.2.1 Application 1

```

from Tkinter import *
class MyEntry:
    def __init__(self,root):
        self.f=Frame(root,height=400, width=600)
        self.f.propagate(0)
        self.f.pack()
        self.l1=Label(text='Enter Your Name') #Create Label name
        self.l2=Label(text='Enter Your age') #Create Label password
        self.l3=Label(text='Select Your City') #Create Label address
        self.l4=Label(text='Select Your Gender') #Create Label Gender
        self.l5=Label(text='Select Favourite games') #Create Label Gender

```

```

        self.e1=Entry(self.f,width=18,fg="black",bg="gray", font=('arial',12))#Add text entry
        widget for name

        self.e2=Entry(self.f,width=18,fg="black",bg="gray", font=('arial',12),show="*") #Add
        text entry widget for address
        self.val1=StringVar()
        self.s1=Spinbox(self.f,values=('Mumbai','Delhi','Kolkata','Chennai'),textvariable=self.v
        al1,width=15,fg='black',bg='gray',font=('Arial',12,'bold italic'))
        self.var=IntVar()
        self.r1=Radiobutton(self.f,text='Male',variable=self.var,value=1,command=self.display
) #Add radiobutton

        self.r2=Radiobutton(self.f,text='Female',variable=self.var,value=2,command=self.displ
ay) #Add radiobutton
        self.var1=IntVar()
        self.var2=IntVar()
        self.var3=IntVar()

        self.c1=Checkbutton(self.f,text='Cricket',variable=self.var1,command=self.display)
#Add Checkbox

        self.c2=Checkbutton(self.f,text='Football',variable=self.var2,command=self.display)
#Add Checkbox

        self.c3=Checkbutton(self.f,text='Tennis',variable=self.var3,command=self.display)
#Add Checkbox

        self.b=Button(self.f,text="Print Details",command=self.display)#Add Button
#Place Labels,entry widget,Buttons,Spinbox on the frame
        self.l1.place(x=50,y=30)
        self.e1.place(x=200,y=30)
        self.l2.place(x=50,y=60)
        self.e2.place(x=200,y=60)
        self.l3.place(x=50,y=90)
        self.s1.place(x=200,y=90)
        self.l4.place(x=50,y=130)
        self.r1.place(x=190,y=130)
        self.r2.place(x=260,y=130)
        self.l5.place(x=50,y=160)
        self.c1.place(x=190,y=160)
        self.c2.place(x=270,y=160)
        self.c3.place(x=360,y=160)

        self.b.place(x=140,y=180)
def display(self):
    str1=self.e1.get()

```

```

str2=self.e2.get()
str3=self.s1.get()
x1=self.var1.get()
y=self.var2.get()
z=self.var3.get()

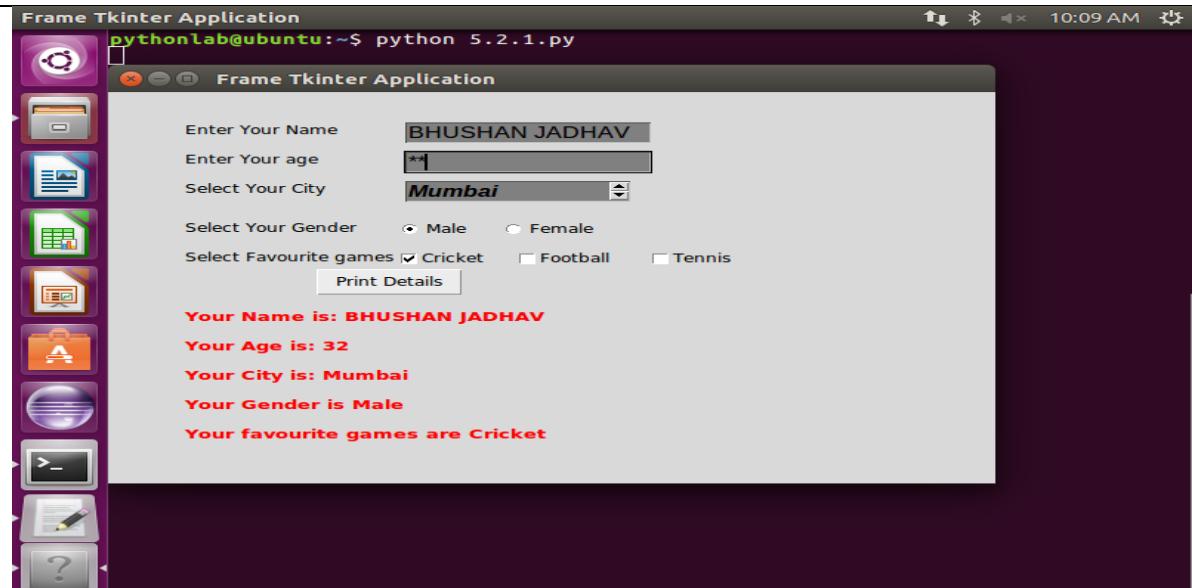
x=self.var.get()
str4=""
if x==1:
    str4+='Your Gender is Male'
if x==2:
    str4+='Your Gender is Female'

str5=""
if x1==1:
    str5+='Cricket'
if y==1:
    str5+='Football'
if z==1:
    str5+='Tennis'

l7=Label(text='Your Name is:
'+str1,font=('Verdana',10,'bold'),fg='red').place(x=50,y=220)#label for entry widget name
l8=Label(text='Your Age is:
'+str2,font=('Verdana',10,'bold'),fg='red').place(x=50,y=250)#label for entry widget age
l9=Label(text='Your City is:
'+str3,font=('Verdana',10,'bold'),fg='red').place(x=50,y=280)#label for Spinbox
l10=Label(text=str4,font=('Verdana',10,'bold'),fg='red').place(x=50,y=310) #label for
radio button
l11=Label(text='Your favourite games are
'+str5,font=('Verdana',10,'bold'),fg='red').place(x=50,y=340) #label for Checkbox

root=Tk()#create root window
root.title("Frame Tkinter Application") #Giving a title to bar
mb=MyEntry(root)#creating an object to class
root.mainloop()#root window handles the mouse events

```



5.2.2 Application 2

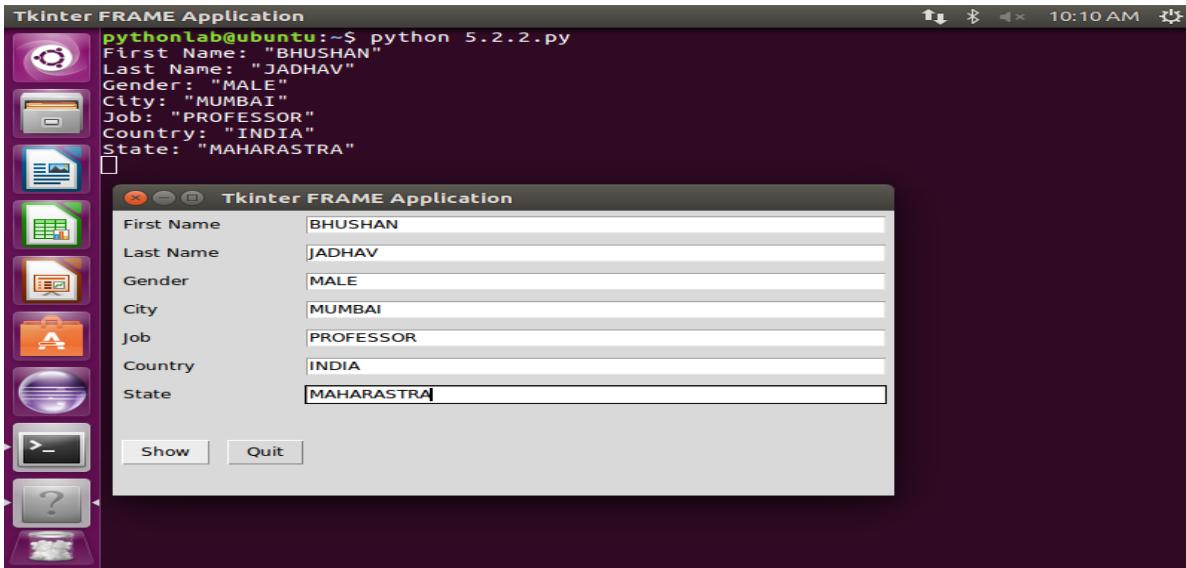
```
from Tkinter import *
fields = 'First Name', 'Last Name', 'Gender','City','Job', 'Country','State',
def fetch(entries):
    for entry in entries:
        field = entry[0]
        text = entry[1].get()
        print('%s: "%s"' % (field, text))

def makeform(root, fields):
    entries = []
    for field in fields:
        row = Frame(root)
        lab = Label(row, width=15, text=field, anchor='w')
        ent = Entry(row)
        row.pack(side=TOP, fill=X, padx=5, pady=5)
        lab.pack(side=LEFT)
        ent.pack(side=RIGHT, expand=YES, fill=X)
        entries.append((field, ent))
    return entries

if __name__ == '__main__':
    root = Tk()
    root.title("Tkinter FRAME Application")
    ents = makeform(root, fields)
    root.bind('<Return>', (lambda event, e=ents: fetch(e)))
    b1 = Button(root, text='Show',
                command=(lambda e=ents: fetch(e)))
```

```
b1.pack(side=LEFT, padx=5, pady=5)
b2 = Button(root, text='Quit', command=root.quit)
b2.pack(side=LEFT, padx=5, pady=5)
root.mainloop()
```

-----**OUTPUT**-----



5.3 Write python programs to understand CRUD Operations using Mysql Python Database Connectivity

Steps to Perform CRUD Operations Python Mysql Database Connectivity

Step 1: Install Mysql in Ubuntu

First you must install a MySQL driver, use the specific installation method below

```
pythonlab@ubuntu:~$ sudo apt-get install mysql-server
```

```
pythonlab@ubuntu:~$ sudo apt-get install python-mysqldb
```

```
pythonlab@ubuntu:~$ sudo apt-get install python-pip python-dev libmysqlclient-dev
```

Step 2: Setup the database

Make sure you have database access, from the command line type:

```
pythonlab@ubuntu:~$ mysql -u root -p
```

MySQL will then ask your password put the password 'python'.

Execute the following commands to create database TESTDB, Create table EMPLOYEE and Insert Records using following commands:

```
mysql> create database TESTDB
```

```
->;
```

```
Query OK, 1 row affected (0.08 sec)
```

```
mysql> USE TESTDB
```

```
Database changed
```

```
mysql> CREATE TABLE EMPLOYEE (
```

```
->      FIRST_NAME CHAR(20) NOT NULL,  
->      LAST_NAME CHAR(20),  
->      AGE INT,  
->      SEX CHAR(1),  
->      INCOME FLOAT )
```

```
->;
```

```
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)  
VALUES ('RAJESH', 'GANGULY', 28, 'M', 93776);
```

```
Query OK, 1 row affected (0.06 sec)
```

```
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)  
VALUES ('SACHIN', 'TENDULKAR', 40, 'M', 98463);
```

```
Query OK, 1 row affected (0.15 sec)
```

```
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)  
VALUES ('ANIL', 'KUMBLE', 42, 'M', 64300);
```

```
Query OK, 1 row affected (0.07 sec)
```

```
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)  
VALUES ('RAHUL', 'DRAVID', 39, 'M', 84300);
```

```
Query OK, 1 row affected (0.07 sec)
```

```
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('YUVRAJ','SINGH',28,'M',94300);
```

Query OK, 1 row affected (0.17 sec)

```
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('VIRAT','KOHLI',26,'M',98343);
```

Query OK, 1 row affected (0.15 sec)

```
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('BHUSHAN','JADHAV',29,'M',94433);
```

Query OK, 1 row affected (0.00 sec)

```
mysql> select * from EMPLOYEE;
```

FIRST_NAME	LAST_NAME	AGE	SEX	INCOME
RAJESH	GANGULY	28	M	93776
SACHIN	TENDULKAR	40	M	98463
ANIL	KUMBLE	42	M	64300
RAHUL	DRAVID	39	M	84300
YUVRAJ	SINGH	28	M	94300
VIRAT	KOHLI	26	M	98343
BHUSHAN	JADHAV	29	M	94433

7 rows in set (0.00 sec)

```
pythonlab@ubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.20-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

```
pythonlab@ubuntu:~$ 
E) VALUES ('VIRAT','KOHLI',26,'M',98343);
Query OK, 1 row affected (0.15 sec)

mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('BHUSHAN','JADHAV',29,'M',94433);
INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME) VALUES ('BHUSHAN','JADHAV',29,'M',94433);
^C
mysql> INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('BHUSHAN','JADHAV',29,'M',94433);
Query OK, 1 row affected (0.00 sec)

mysql> select * from EMPLOYEE;
+-----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+-----+
| RAJESH    | GANGULY   | 28  | M   | 93776  |
| SACHIN    | TENDULKAR | 40  | M   | 98463  |
| ANIL      | KUMBLE    | 42  | M   | 64300  |
| RAHUL     | DRAVID    | 39  | M   | 84300  |
| YUVRAJ    | SINGH     | 28  | M   | 94300  |
| VIRAT     | KOHLI     | 26  | M   | 98343  |
| BHUSHAN   | JADHAV    | 29  | M   | 94433  |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> 
```

Step 2:Run the Python Program

-----Python Program-----

```
#Mysql CRUD Operations using Python

import MySQLdb

# Open database connection

db = MySQLdb.connect("localhost","root","python","TESTDB" )

# prepare a cursor object using cursor() method
```

```
cursor = db.cursor()
print "-----Insert record-----"
# Prepare SQL query to INSERT a record into the database.
a=raw_input('Enter u r First name: ')
b=raw_input('Enter u r Last name: ')
c=int(input('Enter u r Age: '))
d=raw_input('Enter u r Sex: ')
e=int(input('Enter u r Income: '))
sql = "INSERT INTO EMPLOYEE(FIRST_NAME, \
    LAST_NAME, AGE, SEX, INCOME) \
    VALUES ('%s', '%s', '%d', '%c', '%d' )" % \
    (a, b, c, d, e)
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
print "-----View records-----"
sql = "SELECT * FROM EMPLOYEE"
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Fetch all the rows in a list of lists.
    results = cursor.fetchall()
    for row in results:
        fname = row[0]
        lname = row[1]
        age = row[2]
```

```

sex = row[3]
income = row[4]
# Now print fetched result
print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" \% \
    (fname, lname, age, sex, income )

except:
    print "Error: unable to fetch data"
print "-----Update records with age+1 For males-----"
# Prepare SQL query to UPDATE required records
sql = "UPDATE EMPLOYEE SET AGE=AGE+1 WHERE SEX = '%c'" % ('M')
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
print "-----Delete records with age > 30-----"
sql = "DELETE FROM EMPLOYEE WHERE AGE > %d" % (30)
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
print "-----View records-----"
sql = "SELECT * FROM EMPLOYEE"
try:

```

```

# Execute the SQL command
cursor.execute(sql)

# Fetch all the rows in a list of lists.
results = cursor.fetchall()

for row in results:
    fname = row[0]
    lname = row[1]
    age = row[2]
    sex = row[3]
    income = row[4]

    # Now print fetched result
    print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" % \
        (fname, lname, age, sex, income )

except:
    print "Error: unable to fetch data"

# disconnect from server
db.close()

```

-----**OUTPUT**-----

```

pythonlab@ubuntu:~$ python 5.3.py
-----Insert record-----
Enter u r First name: ROHIT
Enter u r Last name: SHARMA
Enter u r Age: 27
Enter u r Sex: M
Enter u r Income: 35432
-----View records-----
fname=RAJESH,lname=GANGULY,age=28,sex=M,income=93776
fname=SACHIN,lname=TENDULKAR,age=40,sex=M,income=98463
fname=ANIL,lname=KUMBLE,age=42,sex=M,income=64300
fname=RAHUL,lname=DRAVID,age=39,sex=M,income=84300
fname=YUVRAJ,lname=SINGH,age=28,sex=M,income=94300

```

```

fname=VIRAT,lname=KOHLI,age=26,sex=M,income=98343
fname=BHUSHAN,lname=JADHAV,age=29,sex=M,income=94433
fname=ROHIT,lname=SHARMA,age=27,sex=M,income=35432
-----Update records with age+1 For males-----
-----Delete records with age > 30-----
-----View records-----
fname=RAJESH,lname=GANGULY,age=29,sex=M,income=93776
fname=YUVRAJ,lname=SINGH,age=29,sex=M,income=94300
fname=VIRAT,lname=KOHLI,age=27,sex=M,income=98343
fname=BHUSHAN,lname=JADHAV,age=30,sex=M,income=94433
fname=ROHIT,lname=SHARMA,age=28,sex=M,income=35432
pythonlab@ubuntu:~$
```

```

pythonlab@ubuntu: ~
pythonlab@ubuntu:~$ python 5.3.py
-----Insert record-----
Enter u r First name: ROHIT
Enter u r Last name: SHARMA
Enter u r Age: 27
Enter u r Sex: M
Enter u r Income: 35432
-----View records-----
fname=RAJESH,lname=GANGULY,age=28,sex=M,income=93776
fname=SACHIN,lname=TENDULKAR,age=40,sex=M,income=98463
fname=ANIL,lname=KUMBLE,age=42,sex=M,income=64300
fname=RAHUL,lname=DRAVID,age=39,sex=M,income=84300
fname=YUVRAJ,lname=SINGH,age=28,sex=M,income=94300
fname=VIRAT,lname=KOHLI,age=26,sex=M,income=98343
fname=BHUSHAN,lname=JADHAV,age=29,sex=M,income=94433
fname=ROHIT,lname=SHARMA,age=27,sex=M,income=35432
-----Update records with age+1 For males-----
-----Delete records with age > 30-----
-----View records-----
fname=RAJESH,lname=GANGULY,age=29,sex=M,income=93776
fname=YUVRAJ,lname=SINGH,age=29,sex=M,income=94300
fname=VIRAT,lname=KOHLI,age=27,sex=M,income=98343
fname=BHUSHAN,lname=JADHAV,age=30,sex=M,income=94433
fname=ROHIT,lname=SHARMA,age=28,sex=M,income=35432
pythonlab@ubuntu:~$
```

Experiment No :- 6

6.1 Write python programs to understand Client-Server programming using TCP Socket

-----tcpserver.py-----

```
import socket
host='127.0.0.1'
port=8000
#Create server side socket
s=socket.socket()
s.bind((host,port))
print "Server is Waiting....."
#Allow max clients connections=1
s.listen(1)
# wait till client connects
c,addr=s.accept()
print "A Client is connected"
#Server runs continuously
while True:
    #receive 1024 byte data from client
    data=c.recv(1024)
    # if client send empty string then come out
    if not data:
        break
    print "From Client: "+str(data.decode())
    # Enter response from server
    data1=raw_input("From Server : ")
    # Send data to the client
    c.send(data1.encode())
#Close Connection
c.close()
```

-----tcpclient.py-----

```
import socket
host='127.0.0.1'
port=8000
#Create client side socket
s=socket.socket()
s.connect((host,port))
# Enter message at Client side
str1=raw_input("Enter your message : ")

# Continue till client exits
while str1!='exit':
    # Send data from client to server
    s.send(str1.encode())
    #receive 1024 byte data from client
    data=s.recv(1024)
    data1=data.decode()
    print "From server :",data1
```

```
#Enter data
str1=raw_input("Enter data : ")

#Close Connection
s.close()
```

-----OUTPUT-----

```
Terminal
pythonlab@ubuntu:~$ python tcpserver.py
Server is Waiting.....
A Client is connected
From Client: Hello
From Server : Welcome
From Client: Doing Python Programming
From Server : Great...
pythonlab@ubuntu:~$ python tcpclient.py
Enter your message : Hello
From server : Welcome
Enter data : Doing Python Programming
From server : Great...
Enter data : ■
```

6.2 Write python programs to understand Client-Server programming using UDP Socket

-----udpserver.py-----

```
import socket
import time
host='localhost'
port=5000
#create Socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
print "Server is Waiting ....."
time.sleep(5)
#Send message to client
s.sendto(b"Hello Client,..How r u ??", (host,port))
msg="Bye"
s.sendto(msg.encode(), (host,port))
s.close()
```

-----udpclient.py-----

```
import socket

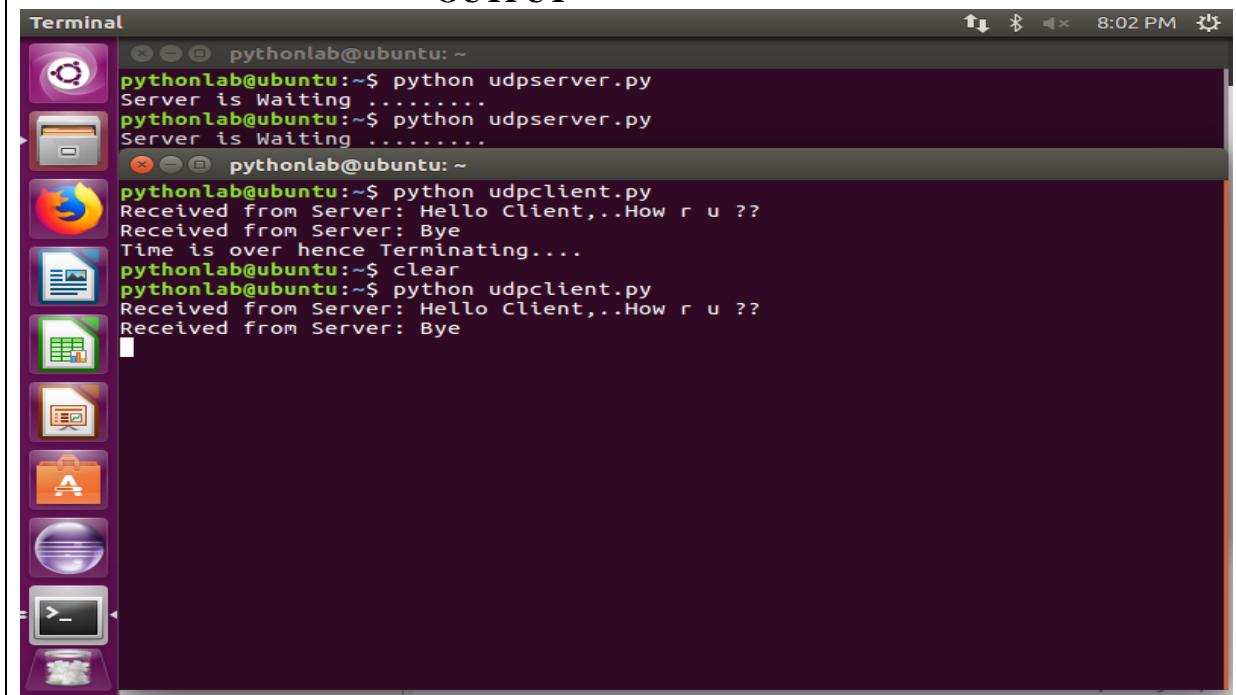
host='localhost'
port=5000
#create client Socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
#Connect to server using hostname and port
```

```

s.bind((host,port))
#receive a message
msg, addr=s.recvfrom(1024)
try:
    #block socket for 5 seconds for sync
    s.settimeout(5)
    #repeat till msg get empty
    while msg:
        print'Received from Server:',msg.decode()
        msg, addr=s.recvfrom(1024)
except socket.timeout:
    print"Time is over hence Terminating...."
s.close()

```

-----OUTPUT-----



The screenshot shows a terminal window on an Ubuntu desktop environment. The terminal window title is "Terminal". The session starts with the command `python udpserver.py`, followed by two identical responses: "Server is Waiting". Then, the command `python udpclient.py` is run, which prints "Received from Server: Hello Client,..How r u ??", "Received from Server: Bye", and "Time is over hence Terminating....". Finally, the command `clear` is run, and then `python udpclient.py` is run again, showing the same exchange of messages.

```

Terminal
pythonlab@ubuntu:~$ python udpserver.py
Server is Waiting .....
pythonlab@ubuntu:~$ python udpserver.py
Server is Waiting .....
pythonlab@ubuntu:~$ python udpclient.py
Received from Server: Hello Client,..How r u ??
Received from Server: Bye
Time is over hence Terminating....
pythonlab@ubuntu:~$ clear
pythonlab@ubuntu:~$ python udpclient.py
Received from Server: Hello Client,..How r u ??
Received from Server: Bye

```

POST LAB EXPERIMENTS

1. Introduction

1. Python Program to Calculate the Area of a Triangle ,square and rectangle
2. Python Program to Solve Quadratic Equations
3. Python Program to Swap Two Variables
4. Python Program to Generate a Random Number
5. Python Program to Convert Kilometers to Miles, Celsius To Fahrenheit and other conversions

2. Decision making and Loops

- 1) Python Program to Check if a Number is Positive, Negative or 0
- 2) Python Program to Check if a Number is Odd or Even
- 3) Python Program to Check Leap Year
- 4) Python Program to Find the Largest Among Three Numbers
- 5) Python Program to Check Prime Number
- 6) Python Program to Print all Prime Numbers in an Interval
- 7) Python Program to Find the Factorial of a Number
- 8) Python Program to Display the multiplication Table
- 9) Python Program to Print the Fibonacci sequence
- 10) Python Program to Check Armstrong Number
- 11) Python Program to Find Armstrong Number in an Interval
- 12) Python Program to Find the Sum of Natural Numbers
- 13) WAP to count number of vowels and consonants from the given strings
- 14) Python program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.
- 15) Python program to demonstrate sorting algorithms using List or Arrays
- 16) Python program to demonstrate searching algorithms using List, Arrays or tuples

3. Functions

- 1) Python Program To Display Powers of 2 Using Anonymous Function
- 2) Python Program to Find Numbers Divisible by Another Number
- 3) Python Program to Convert Decimal to Binary, Octal and Hexadecimal
- 4) Python Program to Find ASCII Value of Character
- 5) Python Program to Find HCF or GCD
- 6) Python Program to Find LCM
- 7) Python Program to Find Factors of Number
- 8) Python Program to Make a Simple Calculator
- 9) Python Program to Shuffle Deck of Cards
- 10) Python Program to Display Calendar
- 11) Python Program to Display Fibonacci Sequence Using Recursion
- 12) Python Program to Find Sum of Natural Numbers Using Recursion
- 13) Python Program to Find Factorial of Number Using Recursion
- 14) Python Program to Convert Decimal to Binary Using Recursion
- 15) Write a Menu driven program in python to implement simple banking application. Application should read the customer name, account number, initial balance, rate of interest, contact number and address field etc. Application should have following methods.
 1. createAccount()

- 2. deposit()
- 3. withdraw()
- 4. computeInterest()
- 5. displayBalance()

16) Write a menu driven code in python which will read a number and should implement the following methods

- 1. factorial()
- 2. reverse()
- 3. testArmstrong()
- 4. testPalindrome()
- 5. testPrime()
- 6. fibonacciSeries()

4. Native Datatypes

- 1) Python Program to Add Two Matrices
- 2) Python Program to Transpose a Matrix
- 3) Python Program to Multiply Two Matrices
- 4) Python Program to Check Whether a String is Palindrome or Not
- 5) Python Program to Remove Punctuations From a String
- 6) Python Program to Sort Words in Alphabetic Order
- 7) Python Program to Illustrate Different Set Operations
- 8) Python Program to Count the Number of Each Vowel

5. File handling

- 1) Python Program to Merge Mails
- 2) Python Program to or delete file ,copy file or move file from one directory in to other directory
- 3) Python Program to Find Hash of File

6. GUI and Database handling using Python

- 1) WAP to draw the house on an canvas using Tkinter.
- 2) On Tkinter Form: Take a Login and Password from the user and display it on the labels which appears only on clicking OK button and clear both the labels on clicking RESET button

- 3) WAP to create a registration form on an Tkinter Form with Mysql Databases.
- 4) To display all the employee records in the emp table
- 5) Update the salary of all the employees in the emp table with an addition of 1000 Rs
- 6) To get the employee id from the user input and display of the details of employees in the emp table
- 7) create the following tables

EMP	DEPT
Eno	Dno
Ename	Dname
Basic	
Hra	

	Da Pf Dno	
a.	Add the records to the above tables	

b. Display a report which shows all the above listed fields grouped by their DNO
 c. Calculate the gross salary of individual and display it.
 d. Calculate department wise total, grand total and display it.

7. Networking

1. WAP to copy a file from client to server using tcp socket
2. WAP for sending a email
3. WAP to find out which of the first 1024 ports seem to be hosting TCP servers
4. WAP to develop chat application
5. WAP that implements socket server to communicate with socket client. Also include logging capability to access pop server

LIST OF MINI PROJECT TOPICS

1. Railway reservation system 2. Inventory Management system. 3 Classroom Management 4 Clinical Trial Initiation and Management 5 Competitive Analysis Web Site 6 Discussion Forum website 7 Disputed Invoice Management 8 Employee Training Scheduling and Materials 9 Equity Research Management 10 Integrated Marketing Campaign Tracking 11 Manufacturing Process Managements 12 Product and Marketing Requirements Planning 13 Request for Proposal Software 14 Sports League Management 15 Absence Request and Vacation Schedule Management 16 Budgeting and Tracking Multiple Projects 17 Bug Database Management 18 Call Center Management Software 19 Change Request Management 20 Compliance Process Support Site 21 Contacts Management Software 22 Document Library and Review 23 Event Planning and Management 24 Expense Reimbursement and Approval 25 Help Desk and Ticket Management 26 Inventory Tracking 27 I T Team Workspace 29 Job Requisition and Interview Management	111.Development of a feature-rich, practical Resource Management System (RMS) 112.Development of a feature rich, practical online Tickets reservation system for Cinema halls. 113.Development of a feature rich, practical Time table generation system for a college. 114.Development of a user friendly ,feature-rich, practical Appraisal Tracker 115.Development of Effort Tracker System 116.Development of a feature-rich, practical ?Web Enabled Estate Agent?(WEEA) 117.Development of a Web Based Mail Client 118.Development of a work flow based Complaint Management System (where the complaints are received through emails) 119.Development of an application for receiving orders for printing digital photographs 120.Development of a work flow based purchase
---	--

<p>28 Knowledge Base</p> <p>29 Lending Library</p> <p>30 Physical Asset Tracking and Management</p> <p>31 Project Tracking Workspace</p> <p>32. Shopping Cart .</p> <p>33 Knowledge Base</p> <p>34 Lending Library</p> <p>35 Physical Asset Tracking and Management</p> <p>36 Project Tracking Workspace</p> <p>37 Room and Equipment Reservations</p> <p>38 Sales Lead Pipeline</p> <p>39. Yellow Pages & Business Directory</p> <p>40. Time & Billing</p> <p>41. Class Room Management</p> <p>42. Expense Report Database</p> <p>43. Sales Contact Management Database</p> <p>44. Inventory Management Database</p> <p>45. Issue Database</p> <p>46. Event Management Database</p> <p>47. Service Call Management Database</p> <p>48. Accounting Ledger Database</p> <p>49. Asset Tracking Database</p> <p>50. Cycle Factory Works Management</p> <p>51. Sales Corporation Management</p> <p>52. Business Directory</p> <p>53. Education Directory</p> <p>54. Dental Clinic Management</p> <p>55. Fund Raising Management</p> <p>56. Clinic/ Health Management</p> <p>57. Cable Management System</p> <p>58. Survey Creation and Analytics</p> <p>59. Museum Management System</p> <p>60. Multi-Level Marketing System</p> <p>61. Learning Management System</p> <p>62. Knowledge Management System</p> <p>63. Missing Person Site</p> <p>64. Disaster Management Site</p> <p>65. Job Management Site</p> <p>66. Financial Portfolio Management</p> <p>67. Market Research Management</p> <p>68. Order Management System</p> <p>69. Point of Sale</p> <p>70. Advertisement /Banner Management and Analytics</p> <p>71. Export Management System</p> <p>72. Invoice Management</p> <p>73. Recruitment Management System</p> <p>74. Articles / Blog / Wiki Web site</p> <p>75. Online Planner</p> <p>76. Mock Tests and Examination Management</p> <p>77. Examination System</p>	<p>request approval system</p> <p>121.Development of a Defect Tracking System (DTS)</p> <p>122.Development of a Product Master Maintenance system</p> <p>123.Development of a Recipe Management System</p> <p>124. Development of a feature-rich Employee Separation System (E-Separation System)</p> <p>125. Development of a Miles Acquisition System (MAS)</p> <p>126. Development of a Network packet sniffer</p> <p>127. Development of a Web Based Meeting Scheduler</p> <p>128. Development of an Employee Cubicle Management System</p> <p>129. Development of a web based Stationery Management System</p> <p>130. Development of an Online Course Portal for a campus</p> <p>131. Development of an Online Auctioning Shop for a campus/organization</p> <p>132. Solving system of linear equations using parallel processing</p> <p>133. Design and development of Point Of Sale [POS]</p> <p>134. Design and development of Speed Cash System [SCS]</p> <p>135. Development of a feature-rich, practical online leave management system (LMS)</p> <p>136. Development of a practical Online Help Desk (OHD) for the facilities in the campus</p> <p>137. Development of an auto-summarization tool</p> <p>138. Development of an agent-based information</p>
--	--

<p>78. Practice Test Management.</p> <p>79. Asset Management System</p> <p>80. Travel Agency System.</p> <p>81. Placement Management System.</p> <p>82. Polls Management</p> <p>83. Customer Management</p> <p>84. Project Management System.</p> <p>85. Network Marketing System</p> <p>86. Yoga Health Care Management</p> <p>87. Personal Finance Management System</p> <p>88. Real Estate Management System</p> <p>89. Stock Mutual Funds Management</p> <p>90. Careers and Employment Management System</p> <p>91. Music Albums Management System</p> <p>92. Classified Ads Managements</p> <p>93. Property Management System</p> <p>94. Sales & Retail Management</p> <p>95. Dating Site</p> <p>96. Hotel Management System</p> <p>97. Search Engine</p> <p>98. Online News Paper Site</p> <p>99. Image Gallery</p> <p>100. Staffing and Human Capital Management</p> <p>101. Development of a feature-rich, practical Online Survey Tool (OST)</p> <p>102 Development of a Web/Email based Search Engine</p> <p>103. Development of a web-based Recruitment Process System for the HR group for a company</p> <p>104. Development of a Budget Approval System</p> <p>105. File system simulation</p> <p>106. Development of a Network Print Spooler</p> <p>107. Development of a HTTP Caching Proxy Server</p> <p>108. Development of a Lost Articles and Letters Reconciliation System</p> <p>109. Student Project Allocation and Management with Online Testing System (SPM)</p> <p>110. Development of a user friendly ,feature-rich, practical Online Testing System (OTS).</p>	<p>push mechanism</p> <p>139. Development of a feature-rich, practical online on-request courses coordination system (ORS)</p> <p>140. Development of an online Library Management System (LiMS)</p> <p>141. Development of an online Sales and Inventory Management System (SIMS)</p> <p>142. Development of a feature-rich, Employee Transfer Application</p> <p>143. Development of a feature-rich, Resume Builder Application</p> <p>144. Development of a safe and secure Internet banking system(Java based) OR Banking System in Visual Basic(Stand Alone)</p> <p>145. Development of a feature-rich, practical online intranet knowledge mgmt system for the college (KMS).</p> <p>146. Development of a feature-rich, practical online application for the Training and Placement Dept. of the college</p> <p>147. Development of a Repository and Search Engine for Alumni of College (RASE)</p> <p>148. Development of a split screen application for the data entry of the shipments.</p> <p>149. Development of a Campaign Information System</p> <p>150 Development of an e-Post Office System</p>
--	--