Python Activity 2 – Solution

```
root@krosumlabs:~# cat -n p1.py
     1  '''
     2  1.Write a python program:
     3  Using range () display CPU load balance 5 times for every 2 seconds delay.
     4  '''
     5  import os
     6  print("Current cpu loadbalance:-")
     7  for v in range(5):
     8          os.system("uptime;sleep 2")
```

Result:-

```
root@krosumlabs:~# python p1.py
 Current cpu loadbalance:-
 17:34:51 up 16:56,  2 users,  load average: 0.00, 0.01, 0.05
 17:34:53 up 16:56,  2 users,  load average: 0.00, 0.01, 0.05
 17:34:55 up 16:56,  2 users,  load average: 0.00, 0.01, 0.05
 17:34:57 up 16:56,  2 users,  load average: 0.00, 0.01, 0.05
 17:34:59 up 16:56,  2 users,  load average: 0.00, 0.01, 0.05
```

```
root@krosumlabs:~# cat -n p2.py
     1
     2  numbers = [16, 50, 300,5, 40, 110]
     3  total=0
     4
     5  for v in numbers:
     6          total=total+v
     7
     8  print("Sum of numbers:{}".format(total))
```

Result:-

```
root@krosumlabs:~# python p2.py
Sum of numbers:521
```

```
root@krosumlabs:~# cat -n p3.py
     1 '''
     2 Write a python program to iterate through the given list.
     3 # Given list -->
       hosts=['host01','host02','host03','host04','host05']
     4
     5 Using membership operator test host03 exists or not
     6 if 'host03' does not exists display suitable message to screen
     7 '''
     8 hosts=['host01','host02','host03','host04','host05']
     9
    10       if("host03" in hosts):
    11       print("host03 is exists")
    12       else:
    13             print("Sorry host03 is not exists")
```

Result:-

```
root@krosumlabs:~# python p3.py
host03 is exists
```

```
root@krosumlabs:~# cat -n p4.py
     1  '''
     2  4. Write a python program
     3  a. Create one empty list
     4  b. Read any 5 file names from keyboard and insert them into
            the list.
     5  c. Display list of files
     6  d. Display each input file details.( use ls -l command)
     7  '''
     8
     9  Fnames=[] # empty list
    10
    11  for v in range(5):
    12          var=raw_input("Enter a filename:")  # 2.x
                # var=input("Enter a filename:")  # 3.x
    13          Fnames.append(var)
    14
    15
    16  print("Input file details:-")
    17  for v in Fnames:
    18          print(v)
    19
    20  print("") # empty line

    21  import os
    22  for v in Fnames:
    23          os.system("ls -l "+v)
```

Result:-

```
root@krosumlabs:~# python p4.py

Enter a filename:p1.py
Enter a filename:p2.py
Enter a filename:/etc/passwd
Enter a filename:p3.py
Enter a filename:/var/log/boot.log
Input file details:-
p1.py
p2.py
/etc/passwd
p3.py
/var/log/boot.log

-rw-r--r-- 1 root root 202 2019-01-18 16:23 p1.py
-rw-r--r-- 1 root root 120 2019-01-18 16:24 p2.py
-rw-r--r-- 1 root root 1678 2014-04-05 09:53 /etc/passwd
-rw-r--r-- 1 root root 402 2019-01-18 16:25 p3.py
-rw-r--r-- 1 root root 1167 2019-01-16 08:43 /var/log/boot.log
```

```
root@krosumlabs:~# cat -n p5.py
     1  '''
     2  Write a python program:
     3  Using List
     4  a. Create 5 different os names.
     5  b. Pint 0th index and 1st index value from list
     6  c. Modify 1st index value
     7  d. Display 1st index value (Compare b statement)
     8  e. Display all the list details.
     9  '''
    10 osnames=["unix","linux","aix","winx","SunOS"]
    11
    12 print(osnames[0])
    13 print(osnames[1])
    14
    15 osnames[1]="ORACLELINUX"
    16 print(osnames[1]) # updated value
    17
    18 print(osnames)
```

Result:-

```
root@krosumlabs:~# python p5.py
unix
linux
ORACLELINUX
['unix', 'ORACLELINUX', 'aix', 'winx', 'SunOS']
```

```
root@krosumlabs:~# cat -n p6.py
     1  '''
     2  6. Write a python program
     3  Using tuple :
     4  a. Using tuple create 5 different servernames.
     5  b. print 0th index and 1st index value from tuple
     6  c. modify 1st index value # Read the error message.
     7  d. display all the tuple details.
     8  '''
     9
    10 servers=("host01.example.com","host02.example.com",
                  "host03.example.com","host04.example.com",
                  "host05.example.com")
    11
    12 print(servers[0])
    13 print(servers[1])
    14
    15 # servers[1]="SERVER"  #  'tuple' object does not support item assignment
    16
    17 print(servers)
```

Result:-

```
root@krosumlabs:~# python p6.py
host01.example.com
host02.example.com

('host01.example.com', 'host02.example.com', 'host03.example.com',
'host04.example.com', 'host05.example.com')
```

```
root@krosumlabs:~# cat -n p7.py
     1  '''
     2  Write a python program
     3  Given List
     4
     5   emp=["ram,sales,pune,1000","kumar,prod,chennai,2000",
          "arun,sales,pune,3000","xerox,HR,mumbai,4500"]

     6  Display employee name, working place and sum of emp's salary
     7  '''

     8  emp=["ram,sales,pune,1000","kumar,prod,chennai,2000","arun,sales,pune,3000",
     9          "xerox,HR,mumbai,4500"]
    10
    11  total=0
    12  for v in emp:
    13          name,dept,place,cost=v.split(",")
    14          print("Emp name:{}\tWorking place is:{}".format(name,place))
    15          total=total+int(cost)
    16
    17
    18  print("*"*50)
    19  print("\t Total Emp's Salary:{}".format(total))
    20  print("*"*50)
```

Result:-

```
root@krosumlabs:~# python p7.py

Emp name:ram  Working place is:pune
Emp name:kumar        Working place is:chennai
Emp name:arun Working place is:pune
Emp name:xerox        Working place is:mumbai
**************************************************
         Total Emp's Salary:10500
**************************************************
```

```
root@krosumlabs:~# cat -n p8.py
     1  '''
     2  Write a python program
     3  Given Tuple :
     4  Conf_files=("/etc/passwd","/etc/pam.d","/etc/hosts")
     5  a. Display total no.of conf files
     6  b. Display 1st and last value
     7  c. Add following files into existing tuple ( "/etc/sysconfig" , "/etc/hostname")
     8  d. Using for loop display Conf_files one by one
     9  '''
    10
    11
    12  Conf_files=("/etc/passwd","/etc/pam.d","/etc/hosts")
    13
    14  print("Total no.of conf files:{}".format(len(Conf_files)))
    15
    16  print("0th index value:{}\t last indexvalue:{}".format(Conf_files[0],Conf_files[-1]))
    17
    18  tv=list(Conf_files) # convert to list
    19  tv.append("/etc/sysconfig")
    20  tv.append("/etc/hostname")
    21  Conf_files=tuple(tv)
    22
    23  print("List of conf files:-")
    24  print("---------------------------------")
    25
    26  for v in Conf_files:
    27          print(v)
    28
    29
    30  print("Total no.of conf files:{}".format(len(Conf_files)))
```

Result:-

```
root@krosumlabs:~# python p8.py
Total no.of conf files:3
0th index value:/etc/passwd      last indexvalue:/etc/hosts
List of conf files:-
---------------------------------
/etc/passwd
/etc/pam.d
/etc/hosts
/etc/sysconfig
/etc/hostname
Total no.of conf files:5
root@krosumlabs:~#
```

```
root@krosumlabs:~# cat -n p9.py
     1  '''
     2  Write a python program
     3  From the given string display "root:bin:x:bash" , "userA:usr:bin:ksh",
     4  "userB:usr:bin:sh"
     5  # use split() function
     6  '''
     7
     8  s1="root:bin:x:bash,userA:usr:bin:ksh,userB:usr:bin:sh"

     9  print(s1.split(","))
    10
    11  s2="root:bin:x:bash,userA:usr:bin:ksh,userB:usr:bin:sh"
    12
    13  print(s2.split(":"))
```

Result:-

```
root@krosumlabs:~# python p9.py

['root:bin:x:bash', 'userA:usr:bin:ksh', 'userB:usr:bin:sh']

['root', 'bin', 'x', 'bash,userA', 'usr', 'bin', 'ksh,userB', 'usr', 'bin', 'sh']
```

root@krosumlabs:~# cat -n p10.py

```
 1  L1=["Line1\n","Line2\n","Line3\n","Line4\n","Line5\n","Line6\n","Line7\n","Line8\n"]
 2
 3  # a. How to display 1st 5line data from given list
 4  print(L1[:5])
 5
 6  # b. Insert "LineXYZ\n" into 2nd index
 7  L1.insert(2,"LineXYZ\n")
 8
 9  # c. Display last 5line elements from given list
10  print(L1[-5:])
11
12  # Convert to tuple and using type() display type
13
14  T=tuple(L1)
15  print("T is belongs to:{}".format(type(T)))
16  print("L1 is belongs to:{}".format(type(L1)))
```

Result:-

root@krosumlabs:~# **python p10.py**

```
['Line1\n', 'Line2\n', 'Line3\n', 'Line4\n', 'Line5\n']
['Line4\n', 'Line5\n', 'Line6\n', 'Line7\n', 'Line8\n']
T is belongs to:<type 'tuple'>
L1 is belongs to:<type 'list'>
root@krosumlabs:~#
```

```
root@krosumlabs:~# cat -n p11.py
     1  '''
     2  Write a python script:
     3  Create an empty list named as "users"
     4  Read 5 emp name from STDIN , insert each name into existing users list
     5  Display all the emp names in sorted order (use : sort function )
     6  '''
     7  users=[]
     8
     9  for v in range(5):
    10          name=raw_input("Enter a user name:")
    11          users.append(name)
    12
    13
    14  print("Sorted order:-")
    15  for v in sorted(users):
    16          print(v)
    17
    18
    19  print(users) # not modified
    20
    21  users.sort() # modified originial structure
    22
    23  for v in users:
    24          print(v)
```

Result:-

```
root@krosumlabs:~# python p11.py

Enter a user name:ram
Enter a user name:xerox
Enter a user name:anu
Enter a user name:yahoo
Enter a user name:paul
Sorted order:-
anu
paul
ram
xerox
yahoo
['ram', 'xerox', 'anu', 'yahoo', 'paul']
anu
paul
ram
xerox
yahoo
root@krosumlabs:~#
```

root@krosumlabs:~# cat -n p12.py

```
     1  '''
     2  Files=("p1.log","p2.log","p3.log","p4.log","p5.log")
     3  Display all the files following format
     4  Output:-
     5  1. p1.log
     6  2. p2.log
     7  3. p3.log
     8  4. p4.log
     9  5. p5.log
    10  Total No.of log files are: 5
    11  '''
    12
    13  Files=("p1.log","p2.log","p3.log","p4.log","p5.log")
    14  c=1
    15  for v in Files:
    16          print("{}. {}".format(c,v))
    17          c=c+1
```

Result:-

```
root@krosumlabs:~# python p12.py
1. p1.log
2. p2.log
3. p3.log
4. p4.log
5. p5.log
root@krosumlabs:~#
```

14.What Is The Output Of The Following Python Code Fragment? Justify Your Answer.

```
>>>
>>> list = ['a', 'b', 'c', 'd', 'e']
>>> list[10:]
[]
>>>
>>> for var in ["mon","tue","wed","thu","fri"]:
...     if(var == "wed"):
...             continue
...     else:
...             print(var)
...
mon
tue
thu
fri
>>>
>>>
>>> Logfiles=("Test1.log","Test2.log","Test3.log","Test4.log")
>>> if("Test2.log" not in Logfiles):
...     print("FOUND")
... else:
...     print("NOT-FOUND")
...
NOT-FOUND
>>>
...
>>> v="root:x:/bin/bash-123:text:bin:text"
>>> v.split("/")[-1]
'bash-123:text:bin:text'
>>>
>>>
>>> weekdays = ['sun','mon','tue','wed','thu','fri','sun','mon','mon']
>>> print(weekdays.count('mon'))
3
>>>
>>>
>>> testList = [1, 3, 5]
>>> testList.sort(reverse=True)
>>> print(testList)
[5, 3, 1]
```

15. Identify the errors in the below codes

```
>>> var=("/etc/passwd","/etc/pam.d","/var/log")
>>>
>>> var.append("/etc/groups)
  File "<stdin>", line 1
    var.append("/etc/groups)
                           ^
SyntaxError: EOL while scanning string literal
>>>
>>> var.append("/etc/groups")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'tuple' object has no attribute 'append'
>>>

>>> cmd=["git","-a","branch","-C"]
>>> cmd[5]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>>


>>> emptylist=[]
>>> emptylist.append("Data1","Data2")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: append() takes exactly one argument (2 given)
>>>


>>> S1="testing data"
>>> del(S1[2])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object doesn't support item deletion
>>>
```

```
>>> "ab" notin "abcd"
  File "<stdin>", line 1
    "ab" notin "abcd"
           ^
SyntaxError: invalid syntax
>>>
>>>

>>> v1=120
>>> v2=input("Enter v2 value:")
Enter v2 value:100
>>>
>>> v1+v2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>>
```

File Handling

```
root@krosumlabs:~# cat -n F1.py
     1  '''
     2  Q1. Write a Python Program read following emp.csv file,
     3  display each employee name,salary,calculate sum of emp's
        salary amount.
     4  emp.csv
     5  --------
     6  arun,sales,1000
     7  vijay,prod,2000
     8  anu,sales,3000
     9  xerox,prod,4000
    10 karthik,HR,5000
    11 '''
    12 total=0
    13 with open("emp.csv") as fobj:
    14         for v in fobj.readlines():
    15                 v=v.strip()
    16                 name,dept,cost=v.split(",")
    17                 print("Emp name:{}\t Salary:{}".format(name,cost))
    18                 total=total+int(cost)
    19
    20
    21
    22 print("----\nSum of emp's salary:{}\n----".format(total))
root@krosumlabs:~#
```

Result:-

```
root@krosumlabs:~# python F1.py
Emp name:arun        Salary:1000
Emp name:vijay       Salary:2000
Emp name:anu  Salary:3000
Emp name:xerox       Salary:4000
Emp name:karthik     Salary:5000
----
Sum of emp's salary:15000
----
root@krosumlabs:~#
```

```
root@krosumlabs:~# cat -n F2.py
     1  '''
     2  Q2.Write a Python program to count the number of lines in a text
     3  file under current directory.
     4  # use directory handling ( import os)
     5  '''
     6
     7  import os
     8  var=os.listdir(".")
     9  print("Total no.of files:{}".format(len(var)))
```

Result:-

```
root@krosumlabs:~# python F2.py
Total no.of files:70
root@krosumlabs:~#
```

root@krosumlabs:~# cat -n F3.py

```
     1  '''
     2  Q3. Write a Python program demonstrate file copy command
            ( cp old new ).
     3  read a data from one file, write a data to another file.
     4  # read input file and result file from STDIN
     5  '''
     6  ifname=raw_input("Enter a input file:")
     7
     8  ofname=raw_input("Enter a result file:")
     9
    10 with open(ifname) as fobj:
    11         with open(ofname,"w") as wobj:
    12                 wobj.write(fobj.read()) # writing data to new file
```

Result:-

root@krosumlabs:~# python F3.py
Enter a input file:**emp.csv**
Enter a result file:**E1.txt**

root@krosumlabs:~# **cat E1.txt**
arun,sales,1000
vijay,prod,2000
anu,sales,3000
xerox,prod,4000
karthik,HR,5000

root@krosumlabs:~# **diff emp.csv E1.txt**

Note : There is no difference emp.csv and E1.txt file

```
root@krosumlabs:~# cat -n F4.py
     1  '''
     2  Q4. Write a python program read below process.log file,
            ignore following shell
     3      names such as "tcsh","p1sh" , "csh" except that the above
     4      shells remaining  shell display to screen.

     5  File: Process.log
     6  sh
     7  bash
     8  tcsh
     9  expect
    10  p1sh
    11  csh
    12  ksh
    13  '''
    14
    15  with open("process.log") as fobj:
    16          for v in fobj.readlines():
    17                  v=v.strip()
    18                  if(v=="tcsh" or v=="p1sh" or v=="csh"):
    19                          continue
    20                  else:
    21                          print(v)
```

Result:-

```
root@krosumlabs:~# python F4.py
sh
bash
expect
ksh
```