Chapter -04 Using Python Libraries

Practical Session -I:

Program 1:

*Create a module **lengthconversion.py** that stores functions for various lengths conversion e.g.,*

- *miletokm() to convert miles to kilometre*
- *kmtomile() to convert kilometres to miles*
- *feettoinches()*
- *inchestofeet()*

It should also store constant values such as value of (mile in kilometres and vice versa).

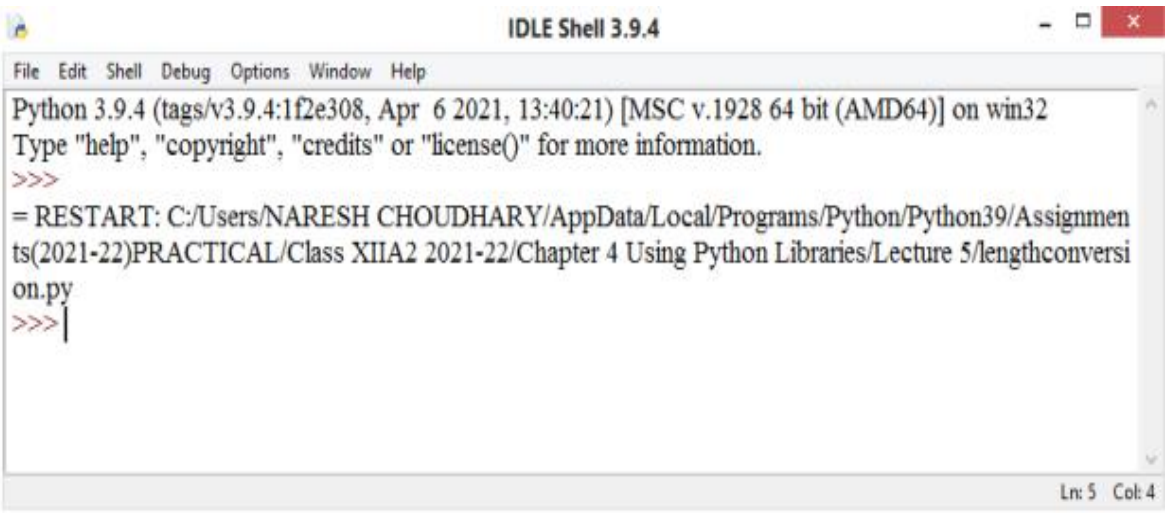
[1 mile =1.609344 kilometre; 1 feet =12 inches]

Help() function should display proper information

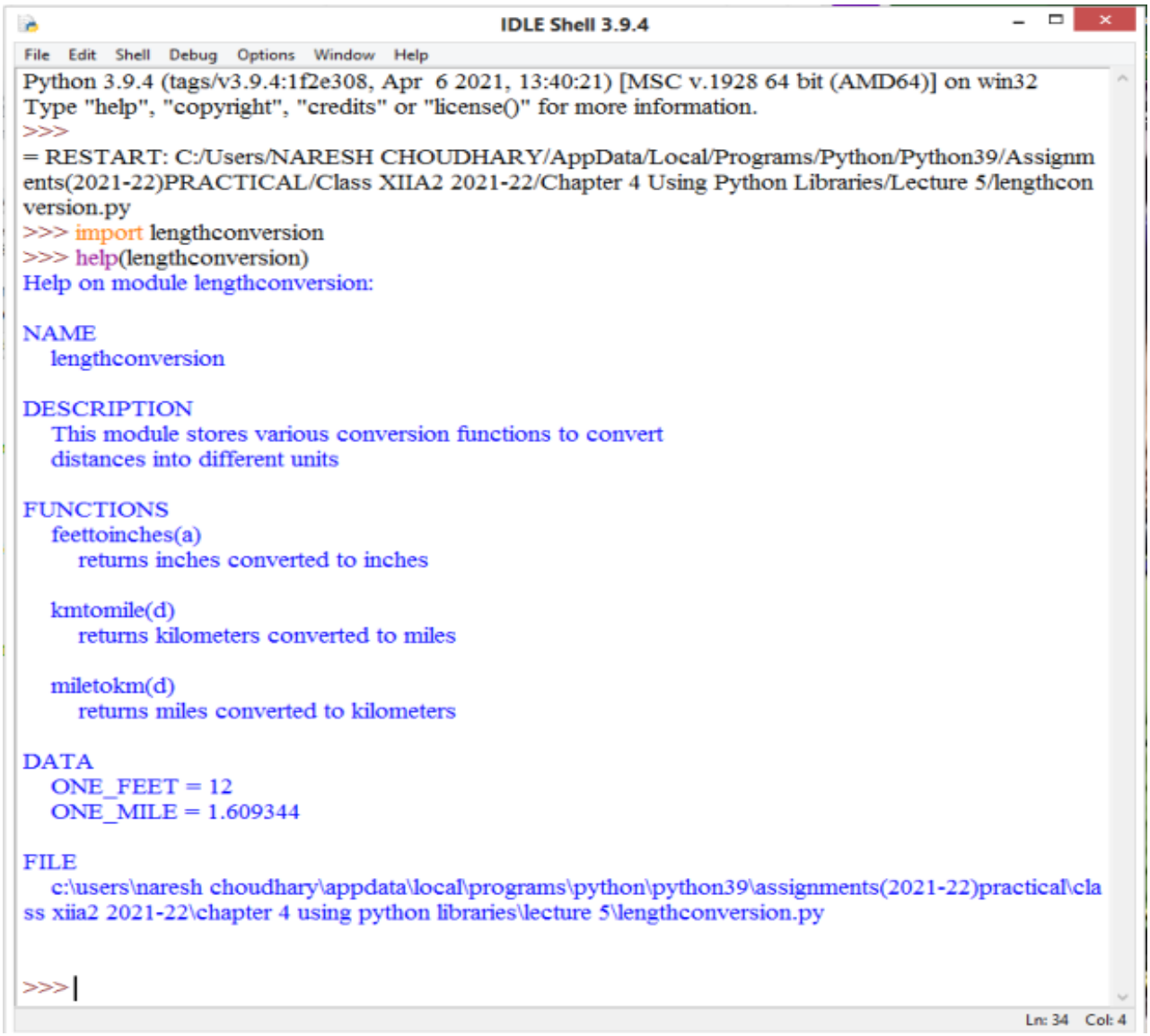
Code:

```
# lengthconversion.py
'''This module stores various conversion functions to convert
distances into different units'''
def miletokm(d):
    #Conversion function 1
    ''' returns miles converted to kilometers'''
    return d * ONE_MILE
def kmtomile(d):
    # Conversion function 2
    ''' returns kilometers converted to miles'''
    return d / ONE_MILE
def feettoinches(a):
    # Conversion function 3
    ''' returns feet converted to inches'''
    return  a * ONE_FEET
def feettoinches(a):
    # Conversion function 4
    ''' returns inches converted to inches'''
    return  a / ONE_FEET

#Constants
ONE_MILE =1.609344              # =1.609
ONE_FEET=12                     # =12 inches
```

Now run the module as:



Now Type help() function/command on the interactive mode:



Program 2:

*Create a **module massconversion.py** that stores function for mass conversion e.g.,*

- *kgtotonne() to convert kg to tonnes*
- *tonnetokg() to convert tonne to kg*
- *kgto pounds() to convert kg to pound*
- *poundtokg() to convert pound to kg*

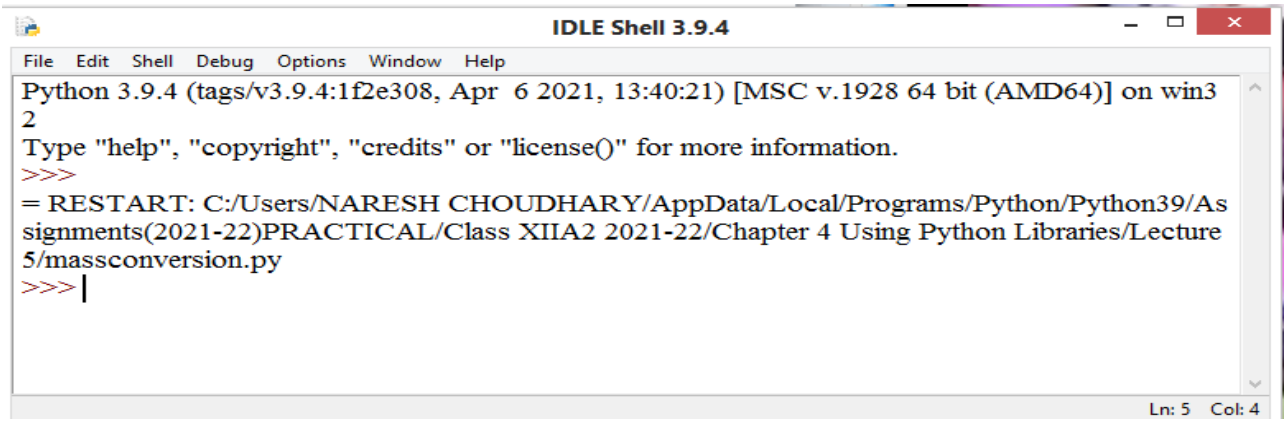
(Also store constants 1 kg =0.001 tonne, 1 kg= 2.20462 pound)

help() function should give proper information about the module

Code:

```
# massconversion.py
'''This module stores various conversion functions to convert
masses into different units'''

def kgtotonne(m):
    #Conversion function 1
    ''' returns kilogramsmiles converted to tonnes'''
    return m * ONE_KG_TONNE

def tonnetokg(m):
    # Conversion function 2
    ''' returns tonnes converted to kilograms'''
    return  m / ONE_KG_TONNE

def kgtopound(a):
    # Conversion function 3
    ''' returns kilograms converted to pounds'''
    return  a * ONE_KG_POUND

def poundtokg(a):
    # Conversion function 4
    ''' returns pounds converted to kilograms'''
    return  a / ONE_KG_POUND
#Constants
ONE_KG_TONNE =0.001        #=0.001 TONNE
ONE_KG_POUND =2.20462      # =2.20462
```

Now run the module massconversion.py as



Now type help () function/command on the interactive mode:

Program 3:

*Create a **module tempcon.py** that stores function for temperature conversion e.g.,*

- *to_centigrade() to convert Fahrenheit to Centigrade*
- *to_fahrenheit() to convert Centigrade to Fahrenheit*

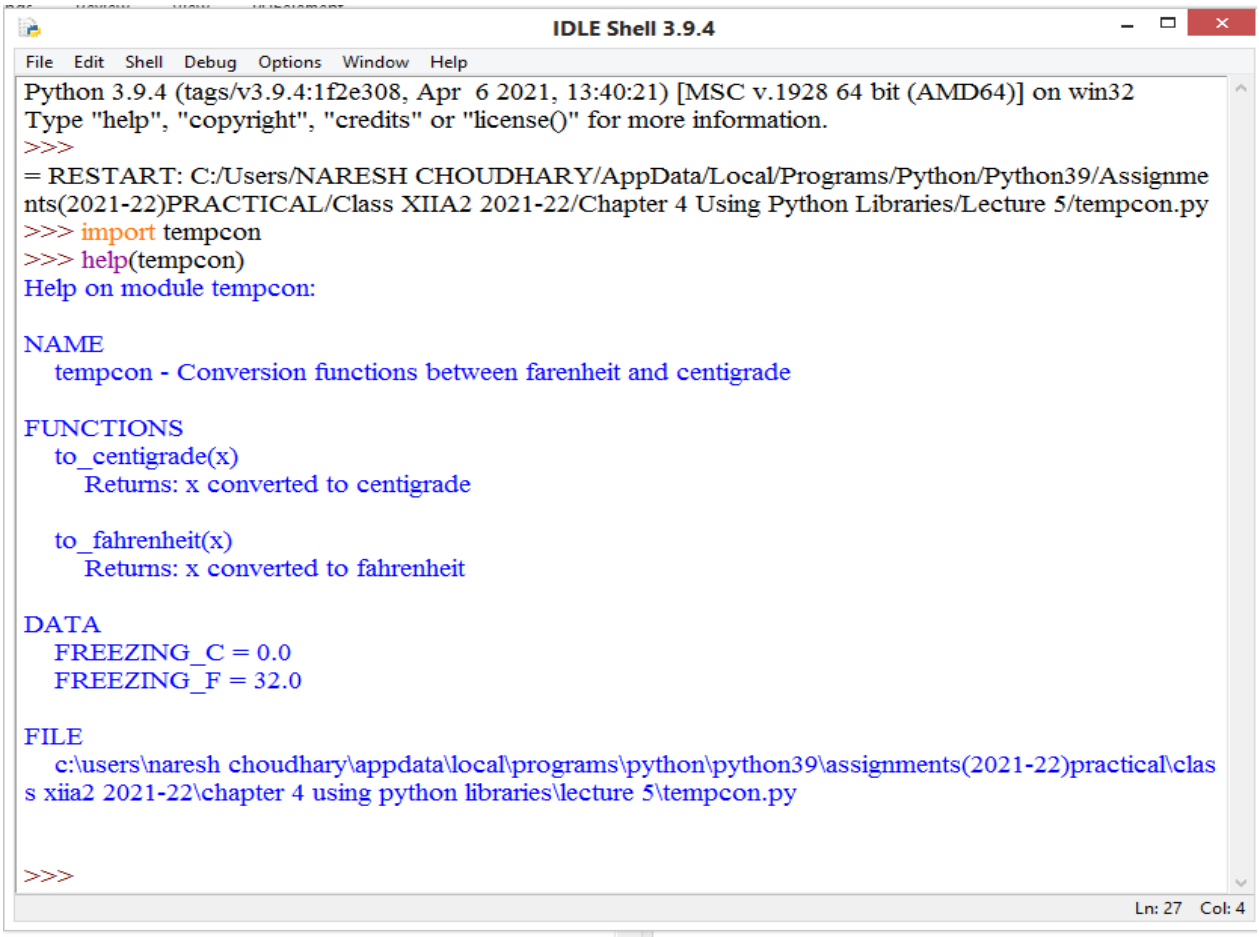
(Also store two constants FREEZING_C =0.0 FREEZING_F =32.0)

help() function should give proper information about the module

Code:

```
#tempcon.py

"""Conversion functions between farenheit and centigrade"""

def to_centigrade(x):
    """Returns: x converted to centigrade """
    return 5 * (x-32)/9.0

def to_fahrenheit(x):
    """Returns: x converted to fahrenheit"""
    return 9 * x /5.0 +32

#Constants
FREEZING_C = 0.0    #Water freezing temp. (in celcius)
FREEZING_F= 32.0    #Water freezing temp. (in fahrenheit)
```

Now run the module tempcon.py and type help() command as



Program 4:

Create a package conversion which include the above two (lengthconversion.py, massconversion.py and temcon.py) modules as like structure given below:

```
Conversion
|
|_______Temperature
|          |___tempconversion.py
|          |___  __init__.py
|
|_____Length
|          |___lengthconversion.py
|          |___ __init__.py
|
|_____Mass
           |___masscon.py
           |___  __init__.py
```

Make sure that above package meets the requirements of being a Python package. Also, program should be able to import above package and and/or its modules using import command.

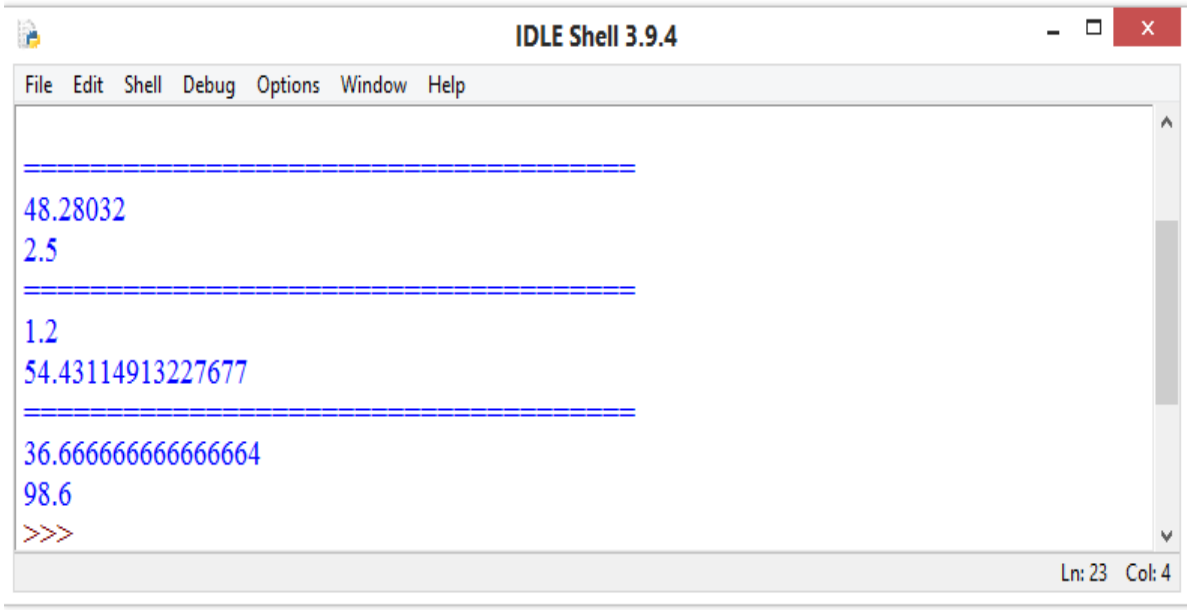
Solution

Structure of *Conversion package* will be displayed as:

Code:-

```
# Call_conversion.py
import Conversion.Length.lengthconversion
import Conversion.Mass.massconversion
import Conversion.Temperature.tempcon
print("\n=====================================")
print(Conversion.Length.lengthconversion.miletokm(30))
print(Conversion.Length.lengthconversion.feettoinches(30))
print("=====================================")

print(Conversion.Mass.massconversion.kgtotonne(1200))
print(Conversion.Mass.massconversion.poundtokg(120))
print("=====================================")

print(Conversion.Temperature.tempcon.to_centigrade(98))
print(Conversion.Temperature.tempcon.to_fahrenheit(37))
```

***Output*:-**


IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
==========
48.28032
2.5
==========
1.2
54.43114913227677
==========
36.666666666666664
98.6
>>>
Ln: 23 Col: 4