

Experiment No:8

Aim: Create reusable modules and packages (e.g., math utilities) and demonstrate the usage of built-in Python modules.

User-Defined Modules

**1. Create a module calc_utils.py with functions for addition, subtraction, multiplication, and division.
Import and use it in another file.**

Code :

```
def add(a,b):
    return a+b
def sub(a,b):
    return a-b
def mul(a,b):
    return a*b
def div(a,b):
    return a/b if b!=0 else 'Inf'
print('Add 2+3 =', add(2,3))
print('Div 10/2 =', div(10,2))
```

Output:

```
Add 2+3 = 5
Div 10/2 = 5.0

==== Code Execution Successful ====
```

2. Develop a module string_ops.py with functions to count vowels, reverse strings, and check for palindromes.

Code :

```
def count_vowels(s):
    return sum(1 for ch in s.lower() if ch in 'aeiou')
def reverse(s):
    return s[::-1]
def is_palindrome(s):
    t="".join(ch.lower() for ch in s if ch.isalnum())
    return t==t[::-1]
print('Vowels in hello:', count_vowels('hello'))
print('Reverse abc:', reverse('abc'))
print('Is Madam palindrome?', is_palindrome('Madam'))
```

Output:

```
Vowels in hello: 2
Reverse abc: cba
Is Madam palindrome? True

==== Code Execution Successful ====
```

3. Create a module list_tools.py to find the largest, smallest, and average of list elements.

Code :

```
def largest(lst):
    return max(lst)
def smallest(lst):
    return min(lst)
def average(lst):
```

Vaishnavi Ashok Patil

C07

SPP-II

```
    return sum(lst)/len(lst)
print('Largest:', largest([1,5,3]))
print('Avg:', average([1,5,3]))
```

Output:

```
Largest: 5
Avg: 3.0

==== Code Execution Successful ====
```

4. Write a module math_ops.py with functions to calculate factorial, square root, and power of a number.

Code :

```
import math
def factorial(n):
    return math.factorial(n)
def sqrt(n):
    return math.sqrt(n)
def power(x,y):
    return x**y
print('Fact 5:', factorial(5))
print('Sqrt 16:', sqrt(16))
```

Output:

```
Fact 5: 120
Sqrt 16: 4.0

==== Code Execution Successful ====
```

5. Create a module temperature.py to convert Celsius ↔ Fahrenheit ↔ Kelvin.

Code :

```
def c_to_f(c):
    return (c*9/5)+32
def f_to_c(f):
    return (f-32)*5/9
def c_to_k(c):
    return c+273.15
print('0C->F', c_to_f(0))
print('32F->C', f_to_c(32))
print('25C->K', c_to_k(25))
```

Output:

```
0C->F 32.0
32F->C 0.0
25C->K 298.15

==== Code Execution Successful ====
```

6. Design a module finance.py to calculate simple interest and compound interest.

Code :

```
def simple_interest(p,r,t):
    return (p*r*t)/100
def compound_interest(p,r,t):
    return p * ((1 + r/100)**t) - p
print('SI for 1000,5%,2yr:', simple_interest(1000,5,2))
print('CI for 1000,5%,2yr:', round(compound_interest(1000,5,2),2))
```

Output:

Vaishnavi Ashok Patil

C07

SPP-II

```
SI for 1000,5%,2yr: 100.0
```

```
CI for 1000,5%,2yr: 102.5
```

```
==> Code Execution Successful ==>
```

7. Create a module geometry.py to calculate area and perimeter of rectangle, triangle, and circle.

Code :

```
import math  
def rect_area(l,w): return l*w  
def rect_perim(l,w): return 2*(l+w)  
def tri_area(b,h): return 0.5*b*h  
def circ_area(r): return math.pi*r*r  
print('Rect area 3x4:', rect_area(3,4))  
print('Circle area r=2:', round(circ_area(2),2))
```

Output:

```
Rect area 3x4: 12  
Circle area r=2: 12.57
```

```
==> Code Execution Successful ==>
```

8. Create a module marks_utils.py to calculate total marks, percentage, and grade of a student.

Code :

```
def total(marks):  
    return sum(marks)  
def percentage(marks):  
    return sum(marks)/len(marks)  
def grade(marks):  
    p=percentage(marks)  
    if p>=90: return 'A'  
    if p>=75: return 'B'  
    if p>=50: return 'C'  
    return 'D'  
marks=[80,85,90]  
print('Total:', total(marks))  
print('Percent:', percentage(marks))  
print('Grade:', grade(marks))
```

Output:

```
Total: 255  
Percent: 85.0  
Grade: B
```

```
==> Code Execution Successful ==>
```

9. Write a module converter.py for unit conversions (cm→m, kg→g, etc.).

Code :

```
def cm_to_m(cm): return cm/100  
def kg_to_g(kg): return kg*1000  
print('200cm->m', cm_to_m(200))  
print('2.5kg->g', kg_to_g(2.5))
```

Output:

Vaishnavi Ashok Patil
C07
SPP-II

```
200cm->m 2.0
2.5kg->g 2500.0
```

```
==> Code Execution Successful ==>
```

10. Create a module date_ops.py to calculate the number of days between two dates.

Code :

```
from datetime import date
def days_between(d1, d2):
    return abs((d2-d1).days)
print('Days between 2020-01-01 and 2020-01-10:', days_between(date(2020,1,1), date(2020,1,10)))
```

Output:

```
Days between 2020-01-01 and 2020-01-10: 9
==> Code Execution Successful ==>
```

11. Write a module string_compare.py to compare two strings lexicographically.

Code :

```
def compare(a,b):
    if a<b: return -1
    if a==b: return 0
    return 1
print('Compare a vs b:', compare('a','b'))
print('Compare abc vs abc:', compare('abc','abc'))
```

Output:

```
Compare a vs b: -1
Compare abc vs abc: 0
==> Code Execution Successful ==>
```

12. Create a module number_utils.py to check whether a number is even, odd, prime, or perfect.

Code :

```
def is_even(n): return n%2==0
def is_prime(n):
    if n<=1: return False
    i=2
    while i*i<=n:
        if n%i==0: return False
        i+=1
    return True
def is_perfect(n):
    return sum(i for i in range(1,n) if n%i==0)==n
print('2 even?', is_even(2))
print('7 prime?', is_prime(7))
print('6 perfect?', is_perfect(6))
```

Output:

```
2 even? True
7 prime? True
6 perfect? True
==> Code Execution Successful ==>
```

13. Build a module file_tools.py with functions to count lines, words, and characters in a text file.

Vaishnavi Ashok Patil

C07

SPP-II

Code :

```
text='Hello world\nThis is a test file\nLine3'
lines = text.splitlines()
words = text.split()
print('Lines:', len(lines))
print('Words:', len(words))
print('Chars:', len(text))
```

Output:

```
Lines: 3
Words: 8
Chars: 37
```

```
==== Code Execution Successful ===
```

14. Create a module matrix_utils.py to perform addition, subtraction, and transpose of 2D matrices.

Code :

```
def add(A,B):
    return [[A[i][j]+B[i][j] for j in range(len(A[0]))] for i in range(len(A))]
def transpose(A):
    return list(map(list, zip(*A)))
A=[[1,2],[3,4]]
B=[[5,6],[7,8]]
print('Add:', add(A,B))
print('Transpose A:', transpose(A))
```

Output:

```
Add: [[6, 8], [10, 12]]
Transpose A: [[1, 3], [2, 4]]
```

```
==== Code Execution Successful ===
```

15. Write a module student_info.py with a function that accepts student details(name, roll no, marks) and displays formatted output.

Code :

```
def display(name, roll, marks):
    print(f'Name: {name} | Roll: {roll} | Marks: {marks} | Total: {sum(marks)}')
display('Ram','C82',[80,85,90])
```

Output:

```
Name: Ram | Roll: C82 | Marks: [80, 85, 90] | Total: 255
```

```
==== Code Execution Successful ===
```

Packages

1. Create a package mathpkg with modules add_sub.py and mul_div.py performing respective operations.

Code :

```
def add(a,b): return a+b
def sub(a,b): return a-b
def mul(a,b): return a*b
def div(a,b): return a/b if b else None
print('add_sub: 5+3=', add(5,3))
print('mul_div: 6*4=', mul(6,4))
```

Output:

Vaishnavi Ashok Patil
C07
SPP-II

```
add_sub: 5+3= 8
mul_div: 6*4= 24

==== Code Execution Successful ====
```

2. Design a package geometry with modules area.py and perimeter.py for basic shapes.

Code :

```
def area_rect(l,w): return l*w
def per_rect(l,w): return 2*(l+w)
print('Area 3x4:', area_rect(3,4))
print('Perim 3x4:', per_rect(3,4))
```

Output:

```
Area 3x4: 12
Perim 3x4: 14

==== Code Execution Successful ====
```

3. Build a package student with modules marks.py (total, percentage) and grade.py (grade evaluation).

Code :

```
def total(marks): return sum(marks)
def percent(marks): return sum(marks)/len(marks)
def grade(p):
    if p>=90: return 'A'
    if p>=75: return 'B'
    return 'C'
m=[80,70,90]
print('Total:', total(m))
print('Percent:', percent(m))
print('Grade:', grade(percent(m)))
```

Output:

```
Total: 240
Percent: 80.0
Grade: B

==== Code Execution Successful ====
```

4. Create a package bank with modules loan.py (EMI calculator) and interest.py (interest calculation).

Code :

```
def emi(p,r,n):
    # r in percent per annum, monthly rate
    r_month = r/(12*100)
    return p * r_month * (1+r_month)**n / ((1+r_month)**n -1)
print('EMI approx:', round(emi(100000,7.5,12),2))
```

Output:

```
EMI approx: 8675.74

==== Code Execution Successful ====
```

5. Design a package stringpkg with modules analysis.py (vowel count, word count) and modify.py (reverse, uppercase).

Code :

```
def vowel_count(s): return sum(1 for ch in s.lower() if ch in 'aeiou')
```

Vaishnavi Ashok Patil

C07

SPP-II

```
def make_upper(s): return s.upper()
print('Vowels in hello:', vowel_count('hello'))
print('Upper:', make_upper('hi'))
```

Output:

```
Vowels in hello: 2
Upper: HI

==== Code Execution Successful ====
```

6. Create a package ecommerce with modules cart.py and billing.py to simulate a simple order system.

Code :

```
cart = []
def add_item(name,price): cart.append((name,price))
def bill():
    return sum(p for _,p in cart)
add_item('pen',10); add_item('book',120)
print('Cart total:', bill())
```

Output:

```
Cart total: 130

==== Code Execution Successful ====
```

7. Develop a package filemanager with modules file_read.py and file_write.py.

Code :

```
fname='/mnt/data/temp_file_exp8.txt'
with open(fname,'w') as f:
    f.write('hello world\nline2')
with open(fname) as f:
    print('File content lines:', f.read().splitlines())
```

Output:

```
File content lines : ['hello world' , 'line2']

==== Code Execution Successful ====
```

8. Create a package converter with modules temperature.py and distance.py (Celsius↔Fahrenheit, km↔miles).

Code :

```
def km_to_miles(km): return km*0.621371
def miles_to_km(mi): return mi/0.621371
print('10km->miles', round(km_to_miles(10),3))
```

Output:

```
10km->miles 6.214

==== Code Execution Successful ====
```

9. Build a package attendance with modules register.py and report.py to mark and display attendance.

Code :

```
register = []
def mark(name): register.append(name)
def report(): print('Present:', register)
mark('A'); mark('B'); report()
```

Output:

Vaishnavi Ashok Patil
C07
SPP-II

```
Present: ['A', 'B']
```

```
==== Code Execution Successful ===
```

10. Create a package games with modules dice.py (random dice roll) and guess.py (number guessing game).

Code :

```
import random
def dice(): return random.randint(1,6)
print('Dice roll:', dice())
```

Output:

```
Dice roll: 1
```

```
==== Code Execution Successful ===
```

11. Develop a package utilities with modules list_utils.py (max, min) and num_utils.py (prime check).

Code :

```
def max_elem(lst): return max(lst)
def is_prime(n):
    if n<=1: return False
    i=2
    while i*i<=n:
        if n%i==0: return False
        i+=1
    return True
print('Max:', max_elem([1,5,3]))
print('7 prime?', is_prime(7))
```

Output:

```
Max: 5
```

```
7 prime? True
```

```
==== Code Execution Successful ===
```

12. Build a package calendarpkg with modules date_info.py and time_info.py.

Code :

```
import datetime
print('Today:', datetime.date.today())
print('Now:', datetime.datetime.now().time())
```

Output:

```
Today: 2025-10-30
```

```
Now: 19:13:37.981423
```

```
==== Code Execution Successful ===
```

13. Create a package travel with modules fare.py and distance.py for basic trip cost calculation.

Code :

```
def fare(dist, rate=10): return dist*rate
def distance(time, speed): return time*speed
print('Fare 5km:', fare(5))
print('Distance 2hr@40:', distance(2,40))
```

Vaishnavi Ashok Patil
C07
SPP-II

Output:

```
Fare 5km: 50
Distance 2hr@40: 80

==== Code Execution Successful ====
```

14. Create a package shop with modules items.py and bill.py to manage product prices and generate bills.

Code :

```
items = {'pen':10,'book':120}
def add_item(name,price): items[name]=price
def total_cart(cart): return sum(items[i] for i in cart)
print('Total cart [pen,book]:', total_cart(['pen','book']))
```

Output:

```
Total cart [pen,book]: 130

==== Code Execution Successful ====
```

15. Design a package school with modules teacher.py (teacher details) and student.py (student details).

Code :

```
def teacher(name,sub): print('Teacher:',name,'Subject:',sub)
def student(name,roll): print('Student:',name,'Roll:',roll)
teacher('T1','Math'); student('S1','C99')
```

Output:

```
Teacher: T1 Subject: Math
Student: S1 Roll: C99

==== Code Execution Successful ====
```

Built-in Python Modules

1. Use the math module to calculate factorial, power, and square root.

Code :

```
import math
print('fact 5:', math.factorial(5))
print('pow 2^3:', math.pow(2,3))
print('sqrt 16:', math.sqrt(16))
```

Output:

```
fact 5: 120
pow 2^3: 8.0
sqrt 16: 4.0

==== Code Execution Successful ====
```

2. Use the math module to find sin, cos, and tan of given angles.

Code :

```
import math
ang = math.radians(45)
print('sin45:', round(math.sin(ang),4))
print('cos45:', round(math.cos(ang),4))
print('tan45:', round(math.tan(ang),4))
```

Output:

Vaishnavi Ashok Patil
C07

SPP-II

```
sin45: 0.7071
cos45: 0.7071
tan45: 1.0
```

```
==== Code Execution Successful ====
```

3. Use the datetime module to display current date and time.

Code :

```
import datetime
print('Now:', datetime.datetime.now())
```

Output:

```
Now: 2025-10-30 19:26:40.443732
```

```
==== Code Execution Successful ====
```

4. Use the datetime module to calculate a person's age from their birth date.

Code :

```
from datetime import date
birth = date(1995,5,15)
now = date.today()
age = now.year - birth.year - ((now.month, now.day) < (birth.month, birth.day))
print('Age:', age)
```

Output:

```
Age: 30
```

```
==== Code Execution Successful ====
```

5. Use the random module to generate random integers between 1 and 100.

Code :

```
import random
print('Random ints:', [random.randint(1,100) for _ in range(5)])
```

Output:

```
Random ints: [82, 43, 18, 56, 98]
```

```
==== Code Execution Successful ====
```

6. Use the random module to simulate rolling two dice and display the result.

Code :

```
import random
print('Dice1, Dice2:', random.randint(1,6), random.randint(1,6))
```

Output:

```
Dice1, Dice2: 6 4
```

```
==== Code Execution Successful ====
```

7. Use the os module to display the current working directory and list all files.

Code :

```
import os
print('CWD:', os.getcwd())
print('Files:', os.listdir('.')[10])
```

Vaishnavi Ashok Patil
C07
SPP-II

Output:

```
CWD: /home/sandbox
Files: ['.ipython', '.openai_internal', 'unicorn_logging.config',
       '.local', '.bashrc', '.config', '.cache', '.profile', '
       .bash_logout']

==== Code Execution Successful ====
```

8. Use the sys module to display Python version and command-line arguments.

Code :

```
import sys
print('Version:', sys.version.split()[0])
print('Argv sample length:', len(sys.argv))
```

Output:

```
Version: 3.11.8
Argv sample length : 3

==== Code Execution Successful ====
```

9. Use the calendar module to print the calendar for a specific year and month.

Code :

```
import calendar
print(calendar.month(2025,10))
```

Output:

```
October 2025
Mo Tu We Th Fr Sa Su
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

==== Code Execution Successful ====
```

10. Use the statistics module to compute mean, median, and mode for a dataset.

Code :

```
import statistics
data=[1,2,2,3,4]
print('Mean:', statistics.mean(data))
print('Median:', statistics.median(data))
print('Mode:', statistics.mode(data))
```

Output:

```
Mean: 2.4
Median: 2
Mode: 2

==== Code Execution Successful ====
```

Vaishnavi Ashok Patil
C07
SPP-II

11. Use the time module to measure how long it takes to execute a loop.

Code :

```
import time
start=time.time()
for i in range(100000): pass
end=time.time()
print('Elapsed (ms):', (end-start)*1000)
```

Output:

```
Elapsed (ms): 4.112958908081055
==> Code Execution Successful ==>
```

12. Use the platform module to display system and OS information.

Code :

```
import platform
print('Platform:', platform.system(), platform.release())
```

Output:

```
Platform: Linux 6.6.97+
==> Code Execution Successful ==>
```

13. Use the json module to read and write data to a JSON file.

Code :

```
import json
obj={'a':1,'b':2}
path='/mnt/data/exp8_sample.json'
with open(path,'w') as f: json.dump(obj,f)
with open(path) as f: print('Loaded:', json.load(f))
```

Output:

```
Loaded: {'a' : 1 , 'b' : 2}
==> Code Execution Successful ==>
```

14. Use the csv module to read data from a CSV file and display it.

Code :

```
import csv
path='/mnt/data/exp8_sample.csv'
with open(path,'w',newline='') as f:
    w=csv.writer(f); w.writerow(['name','age']); w.writerow(['A',20])
with open(path) as f:
    r=csv.reader(f); print('Rows:', list(r))
```

Output:

```
Rows:[['name' , 'age'] , ['A' , '20']]
==> Code Execution Successful ==>
```

15. Use the collections module to count word frequency in a paragraph using Counter().

Vaishnavi Ashok Patil

C07

SPP-II

Code :

```
from collections import Counter
text='apple banana apple orange banana apple'
print(Counter(text.split()))
```

Output:

```
Counter({'apple' : 3, 'bannana': 2 , 'orange':1})
```

```
==== Code Execution Successful ===
```