

Experiment 8 - Reusable Modules, Packages, and Built-in Python Modules

Name: Mohammad Sayeed

Roll no : C56 Div: C Class: TY CSE

1. calc_utils module

Section: User-Defined Modules

Problem Statement:

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 None

print('add(10,5) ->', add(10,5))
print('sub(10,5) ->', sub(10,5))
print('mul(10,5) ->', mul(10,5))
print('div(10,5) ->', div(10,5))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
add(10,5) -> 15
sub(10,5) -> 5
mul(10,5) -> 50
div(10,5) -> 2.0
```

2. string_ops module

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def count_vowels(s): print(sum(1 for ch in s.lower() if ch in 'aeiou'))
def reverse_str(s): print(s[::-1])
def is_pal(s): print('Yes' if ''.join(c.lower() for c in s if c.isalnum())==''.join(c.lower() for c in s if c.isalnum())[::-1] else 'No')

count_vowels('Hello World')
reverse_str('Python')
is_pal('Madam')
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
3
nohtyP
Yes
```

[3. list_tools module](#)

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def largest(lst): print(max(lst))
def smallest(lst): print(min(lst))
def average(lst): print(sum(lst)/len(lst))

lst=[3,7,1,9,4]
largest(lst); smallest(lst); average(lst)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
9
1
4.8
```

[4. math_ops module](#)

Section: User-Defined Modules

Problem Statement:

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

Code:

```
import math
def fact(n):
    res=1
    for i in range(2,n+1): res*=i
    print(res)
def sqrt(n): print(math.sqrt(n))
def power(x,y): print(x**y)

fact(5); sqrt(16); power(2,8)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
120
4.0
256
```

[5. temperature module](#)

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def c_to_f(c): print((c*9/5)+32)
def f_to_c(f): print((f-32)*5/9)
def c_to_k(c): print(c+273.15)

c_to_f(25); f_to_c(77); c_to_k(25)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
77.0
25.0
298.15
```

6. finance module

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def simple_interest(p,r,t): print((p*r*t)/100)

def compound_interest(p,r,t): print(p * ((1 + r/100)**t - 1))

simple_interest(1000,5,2)

compound_interest(1000,5,2)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
100.0
102.5
```

7. geometry module

Section: User-Defined Modules

Problem Statement:

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

Code:

```
import math

def rect_area(l,w): print(l*w)
def rect_perim(l,w): print(2*(l+w))
def tri_area(b,h): print(0.5*b*h)
def circ_area(r): print(math.pi*r*r)

rect_area(4,5); rect_perim(4,5); tri_area(6,4); circ_area(3)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
20
18
12.0
28.274333882308138
```

8. marks_utils module

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def total(marks): print(sum(marks))
def percentage(marks): print(sum(marks)/len(marks))
def grade(marks):
    p=sum(marks)/len(marks)
    if p>=75: g='A'
    elif p>=60: g='B'
    elif p>=50: g='C'
    else: g='D'
    print(g)

marks=[78,82,90,67,71]
total(marks); percentage(marks); grade(marks)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
388
77.6
A
```

[9. converter module](#)

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def cm_to_m(cm): print(cm/100)
def kg_to_g(kg): print(kg*1000)
cm_to_m(250); kg_to_g(5.5)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
2.5
5500.0
```

[10. date_ops module](#)

Section: User-Defined Modules

Problem Statement:

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

Code:

```
from datetime import datetime
d1=datetime(2023,1,1); d2=datetime(2023,12,31)
print((d2-d1).days)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
364
```

11. string_compare module

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def compare(a,b):
    if a<b: print(f"'{a}' < '{b}'")
    elif a>b: print(f"'{a}' > '{b}'")
    else: print("Equal")

compare('apple','banana'); compare('cat','car')
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
'apple' < 'banana'
'cat' > 'car'
```

12. number_utils module

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def even_odd(n): print('Even' if n%2==0 else 'Odd')
def is_prime(n):
    if n<=1: print('No'); return
    for i in range(2,int(n**0.5)+1):
        if n%i==0: print('No'); return
    print('Yes')
def is_perfect(n):
    s=sum(i for i in range(1,n) if n%i==0)
    print('Yes' if s==n else 'No')

even_odd(28); is_prime(29); is_perfect(28)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
Even
Yes
Yes
```

[13. file_tools module](#)

Section: User-Defined Modules

Problem Statement:

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

Code:

```
text='Hello world\nThis is a test\nLine three'
print(len(text.splitlines()))
print(sum(len(line.split()) for line in text.splitlines()))
print(len(text))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
3
8
37
```

[14. matrix_utils module](#)

Section: User-Defined Modules

Problem Statement:

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

Code:

```
A=[[1,2],[3,4]]; B=[[5,6],[7,8]]
add=[[A[i][j]+B[i][j] for j in range(2)] for i in range(2)]
sub=[[A[i][j]-B[i][j] for j in range(2)] for i in range(2)]
trans=[[A[j][i] for j in range(2)] for i in range(2)]
print(add); print(sub); print(trans)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[6, 8], [10, 12]]
[[-4, -4], [-4, -4]]
[[1, 3], [2, 4]]
```

15. student_info module

Section: User-Defined Modules

Problem Statement:

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

Code:

```
def show(name,roll,marks):
    print(f"Name: {name}\nRoll: {roll}\nMarks: {marks}")
show('Rahul', '101', [78,82,90])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
Name: Rahul
Roll: 101
Marks: [78, 82, 90]
```

16. mathpkg add_sub and mul_div

Section: Packages

Problem Statement:

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(5,3), sub(5,3), mul(5,3), div(6,3))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
8 2 15 2.0
```

17. geometry package area and perimeter

Section: Packages

Problem Statement:

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

Code:

```
import math  
def area_rect(l,w): print(l*w)  
def peri_rect(l,w): print(2*(l+w))  
area_rect(4,5); peri_rect(4,5)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
20  
18
```

18. student package marks and grade

Section: Packages

Problem Statement:

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

Code:

```
def total(m): print(sum(m))  
def percentage(m): print(sum(m)/len(m))  
def grade(m):  
    p=sum(m)/len(m)  
    if p>=75: print('A')  
    elif p>=60: print('B')  
    else: print('C')  
m=[70,80,90]; total(m); percentage(m); grade(m)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
240
80.0
A
```

19. bank package loan and interest

Section: Packages

Problem Statement:

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

Code:

```
def emi(p,r,n):
    r=r/100/12
    emi = p*r*(1+r)**n/((1+r)**n-1)
    print(round(emi,2))
emi(100000,10,12)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
8791.59
```

20. stringpkg analysis and modify

Section: Packages

Problem Statement:

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

Code:

```
s='Hello world'
print(sum(1 for c in s.lower() if c in 'aeiou'))
print(len(s.split()))
print(s[::-1].upper())
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
3
2
DLROW OLLEH
```

21. ecommerce cart and billing

Section: Packages

Problem Statement:

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

Code:

```
cart=[('apple',2,30),('bread',1,40)]
total=sum(q*p for _,q,p in cart)
print(total)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
100
```

22. filemanager file_read and file_write

Section: Packages

Problem Statement:

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

Code:

```
text='Sample text for file manager'
print(text.upper())
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
SAMPLE TEXT FOR FILE MANAGER
```

23. converter package temperature and distance

Section: Packages

Problem Statement:

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

Code:

```
def c_to_f(c): print((c*9/5)+32)
def km_to_miles(k): print(k*0.621371)
c_to_f(25); km_to_miles(5)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
77.0
3.106855
```

24. attendance package register and report

Section: Packages

Problem Statement:

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

Code:

```
att={'A':1,'B':0,'C':1}
print(sum(att.values()), 'present')
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
2 present
```

25. games package dice and guess

Section: Packages

Problem Statement:

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

Code:

```
import random  
print(random.randint(1, 6))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
1
```

26. utilities list_utils and num_utils

Section: Packages

Problem Statement:

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

Code:

```
lst=[4,2,7,1]  
print(max(lst), min(lst))  
def is_prime(n):  
    if n<=1: print('No'); return  
    for i in range(2,int(n**0.5)+1):  
        if n%i==0: print('No'); return  
    print('Yes')  
is_prime(17)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
7 1  
Yes
```

27. calendarpkg date_info and time_info

Section: Packages

Problem Statement:

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

Code:

```
import datetime
print(datetime.date.today())
print(datetime.datetime.now().time())
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
2025-11-01
05:28:18.355547
```

28. travel package fare and distance

Section: Packages

Problem Statement:

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

Code:

```
def fare(dist,rate): print(dist*rate)
fare(10,5)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
50
```

29. shop package items and bill

Section: Packages

Problem Statement:

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

Code:

```
items={'apple':30,'bread':40}
total=sum(items.values())
print(total)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
70
```

30. school package teacher and student

Section: Packages

Problem Statement:

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

Code:

```
print('Teacher: Mr. X')  
print('Student: Rahul')
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
Teacher: Mr. X  
Student: Rahul
```

31. math factorial power sqrt

Section: Built-in Modules

Problem Statement:

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

Code:

```
import math  
print(math.factorial(5), math.pow(2,8), math.sqrt(16))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
120 256.0 4.0
```

32. math trig functions

Section: Built-in Modules

Problem Statement:

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

Code:

```
import math
print(math.sin(math.radians(30)), math.cos(math.radians(60)),
math.tan(math.radians(45)))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
0.4999999999999994 0.5000000000000001 0.9999999999999999
```

33. datetime current date and time

Section: Built-in Modules

Problem Statement:

Use the datetime module to display current date and time.

Code:

```
import datetime
print(datetime.datetime.now())
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
2025-11-01 05:28:18.437396
```

34. datetime calculate age

Section: Built-in Modules

Problem Statement:

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

Code:

```
from datetime import date
birth = date(2000,5,15)
today = date.today()
age = today.year - birth.year - ((today.month, today.day) <
```

```
(birth.month, birth.day))  
print(age)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
25
```

35. random integers

Section: Built-in Modules

Problem Statement:

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

Code:

```
import random  
print(random.randint(1,100))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
12
```

36. random two dice

Section: Built-in Modules

Problem Statement:

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

Code:

```
import random  
print(random.randint(1,6), random.randint(1,6))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
1 6
```

[37. os cwd and list files](#)

Section: Built-in Modules

Problem Statement:

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

Code:

```
import os
print(os.getcwd())
print(len(os.listdir('.')))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
/home/sandbox
9
```

[38. sys version and args](#)

Section: Built-in Modules

Problem Statement:

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

Code:

```
import sys
print(sys.version.split()[0])
print(sys.argv[:])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
3.11.8
[/home/sandbox/.local/lib/python3.11/site-packages/ipykernel_launcher.py', '-f', '/tmp/tmpzj822wuw.json']
```

[39. calendar print month](#)

Section: Built-in Modules

Problem Statement:

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

Code:

```
import calendar
print(calendar.month(2025,11))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
November 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
```

40. statistics mean median mode

Section: Built-in Modules

Problem Statement:

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

Code:

```
import statistics
data=[1,2,2,3,4]
print(statistics.mean(data), statistics.median(data),
statistics.mode(data))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
2.4 2 2
```

41. time measure loop

Section: Built-in Modules

Problem Statement:

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

Code:

```
import time
t=time.time()
```

```
for i in range(100000): a=i*i  
print(round(time.time()-t,4))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
0.0714
```

42. platform system info

Section: Built-in Modules

Problem Statement:

Use the platform module to display system and OS information.

Code:

```
import platform  
print(platform.system(), platform.release())
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
Linux 4.4.0
```

43. json read write

Section: Built-in Modules

Problem Statement:

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

Code:

```
import json  
data={'a':1,'b':2}  
s=json.dumps(data)  
print(s)  
print(json.loads(s))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
{"a": 1, "b": 2}  
{'a': 1, 'b': 2}
```

44. csv read

Section: Built-in Modules

Problem Statement:

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

Code:

```
import csv, io
s='name,age\nAlice,23\nBob,30'
f=io.StringIO(s)
reader=csv.reader(f)
for row in reader: print(row)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
['name', 'age']
['Alice', '23']
['Bob', '30']
```

45. collections Counter

Section: Built-in Modules

Problem Statement:

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

Code:

```
from collections import Counter
p='hello world hello'
print(Counter(p.split()))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
Counter({'hello': 2, 'world': 1})
```