8. Python Module & Package

a) Write a program to determine whether a given year is a leap year [Use Calendar Module].

```
import calendar

year = int(input("Enter the year: "))

# Check if it is a leap year
leap_year = calendar.isleap(year)

if leap_year:
    print("The given year is a leap year")
else:
    print("The given year is a non-leap year")
```

b) Write a python script to display

i. Current date and time
ii. Current Year
iii. Month of the year
iv. Week number of the year
v. Weekday of the week
vi. Day of year
vii. Day of the month
viii. Day of week [Use time and datetime Module]

```
import time
import datetime
# Imports the datetime module which allows working with
dates and times

print("current date and time:", datetime.datetime.now())
# Displays the current date and time (year, month, day,
hour, minute, second)

print("current year:",
datetime.date.today().strftime("%Y"))
# Displays the current year (e.g., 2025)

print("month of year:",
datetime.date.today().strftime("%B"))
# Displays the full name of the current month (e.g.,
November)

print("week number of the year",
datetime.date.today().strftime("%W"))
# Displays the week number of the year (from 00 to 53)
```

```
print("Day number of the week:",
datetime.date.today().strftime("%w"))
# Displays the day number of the week (Sunday = 0, Monday =
1, … Saturday = 6)

print("day of year:", datetime.date.today().strftime("%j"))
# Displays the day number of the year (from 001 to 366)

print("day of the month:",
datetime.date.today().strftime("%d"))
# Displays the day of the month (from 01 to 31)

print("day of week", datetime.date.today().strftime("%A"))
# Displays the full name of the day (e.g., Tuesday)

# Using time module
print("\nUsing time module:")
current_time = time.localtime()
print("Today's date:", time.strftime("%Y-%m-%d",
current_time))
```

c) Write a function in file Armstrong.py to check whether a number is an Armstrong number. Import the module to generate Armstrong numbers between two limits.

Module Armstrong.py

```
def armstrong_number(num):
    # Find the number of digits
    order = len(str(num))

    # Temporary variable to process each digit
    temp = num
    sum = 0

    # Loop to calculate sum of digits raised to the power
of 'order'
    while temp > 0:
        digit = temp % 10  # Extract last digit
        sum += digit ** order  # Add digit raised to power
of number of digits
        temp //= 10  # Remove last digit

    # Return True if Armstrong number, otherwise False
    return num == sum
```

Main.py

```
from Armstrong import armstrong_number

number = int(input("Enter a number to check if it's an
Armstrong number: "))

if armstrong_number(number):
    print(f"{number} is an Armstrong number.")
else:
    print(f"{number} is not an Armstrong number.")
```

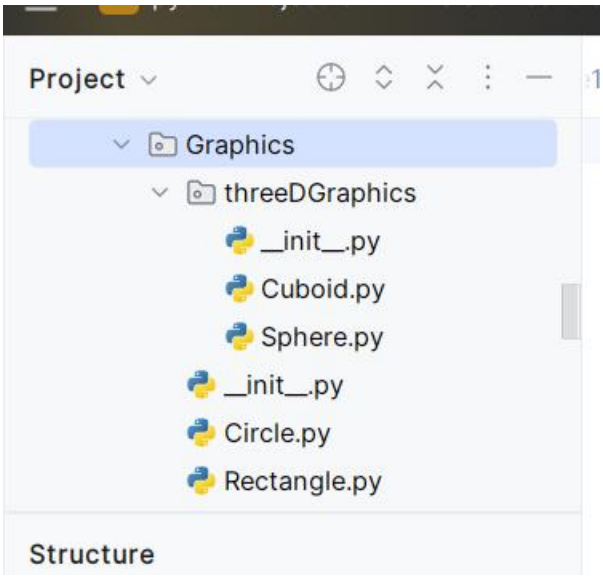
d) Create a package graphics with modules rectangle, circle and sub-package threeDgraphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that find the area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

File - > New - > Create Python Package - Graphics

Graphics -> Create two python files Rectangle.py and Circle.py

Graphics -> Create sub-package threeDGraphics

threeDGraphics - >Create two Python Files Cuboid.py and Sphere.py



```
import Graphics
from Graphics import Circle, Rectangle
from Graphics.threeDGraphics import Cuboid, Sphere
from Graphics.Circle import *

# Circle operations
rad1 = float(input("Enter the radius of the circle: "))
print("Area of circle is:", Circle.area_circle(rad1))
print("Perimeter of circle is:",
Circle.perimeter_circle(rad1))

# Rectangle operations
len1 = float(input("Enter the length of the rectangle: "))
breadth1 = float(input("Enter the breadth of the rectangle:
"))
print("Area of rectangle is:", Rectangle.area_rec(len1,
breadth1))
print("Perimeter of rectangle is:",
Rectangle.perimeter_rec(len1, breadth1))

# Cuboid operations
len2 = float(input("Enter the length of the cuboid: "))
breadth2 = float(input("Enter the breadth of the cuboid: "))
height = float(input("Enter the height of the cuboid: "))
print("Area of cuboid is:", Cuboid.area_cuboid(len2, breadth2,
height))
print("Perimeter of cuboid is:", Cuboid.perimeter_cuboid(len2,
breadth2, height))

# Sphere operations
rad2 = float(input("Enter the radius of the sphere: "))
print("Area of sphere is:",Sphere.area_sphere(rad2))
print("Perimeter of sphere is:",
Sphere.perimeter_sphere(rad2))
```

circle.py

```
from math import pi

# Function to find area of a circle
def area_circle(radius):
    return pi * radius * radius

# Function to find perimeter (circumference) of a circle
def perimeter_circle(radius):
    return 2 * pi * radius
```

rectangle.py

```
def area_rec(length,width):
    return length*width
def perimeter_rec(length,width):
    return 2*(length+width)
```

cuboid.py

```
# Function to find the surface area of a cuboid
def area_cuboid(l, b, h):
    return 2 * (l*b + b*h + l*h)

# Function to find the perimeter of a cuboid (sum of all
edges)
def perimeter_cuboid(l, b, h):
    return 4 * (l + b + h)
```

sphere.py

```
from math import pi

# Function to find the surface area of a sphere
def area_sphere(radius):
    return 4 * pi * radius * radius

# Function to find the perimeter (circumference of great
circle) of a sphere
def perimeter_sphere(radius):
    return 2 * pi * radius
```