# Lab Cycle 5

## Experiment 1

**Aim:** Write a program to determine whether a given year is a leap year [ Use Calendar Module]

## Source Code

```
import calendar
yrs=int(input("enter a year: "))
if calendar.isleap(yrs):
    print(f"{yrs} is a leap year.")
else:
    print(f"{yrs} is not a leap year.")
```

## Output

```
24mca8@softlab-ThinkCentre-M92p:~/pylab/cycle5$ nano 5exp1.py
24mca8@softlab-ThinkCentre-M92p:~/pylab/cycle5$ python3 5exp1.py
enter a year: 2024
2024 is a leap year.
24mca8@softlab-ThinkCentre-M92p:~/pylab/cycle5$ python3 5exp1.py
enter a year: 2023
2023 is not a leap year.
```

## Experiment 2

**Aim:** Write a python script to display
a) Current date and time
b) Current Year
c) Month of the year
d) Week number of the year
e) Weekday of the week
f) Day of year
g) Day of the month
h) Day of week [ Use time and datetime Module]

## Source Code

```
from datetime import datetime
print("current date and time: ",datetime.now())
print("current year: ",datetime.now().year)
print("month of the year: ",datetime.now().strftime("%B"))
print("week no. of year: ",datetime.now().isocalendar()[1])
print("week day of year: ",datetime.now().strftime("%A"))
print("day of year: ",datetime.now().timetuple().tm_yday)
print("day of month: ",datetime.now().day)
print("day of week: ",datetime.now().weekday())
```

## Output

```
24mca8@softlab-ThinkCentre-M92p:~/pylab/cycle5$ python3 5exp2.py
current date and time:  2024-12-10 00:31:41.776417
current year:  2024
month of the year:  December
week no. of year:  50
week day of year:  Tuesday
day of year:  345
day of month:  10
day of week:  1
```

## Experiment 3

**Aim:**Write a python program to print yesterday, today and tomorrow.

## Source Code

```
import datetime
today = datetime.date.today()
yesterday = today - datetime.timedelta(days=1)
tomorrow = today + datetime.timedelta(days=1)
print(f"Yesterday: {yesterday}")
print(f"Today: {today}")
print(f"Tomorrow: {tomorrow}")
```

## Output

```
24mca8@softlab-ThinkCentre-M92p:~/pylab/cycle5$ python3 5exp3.py
Yesterday: 2024-12-09
Today: 2024-12-10
Tomorrow: 2024-12-11
```

## Experiment 4

**Aim:**Write a function in file palindrome.py to check whether a string is Palindrome or not. Import the module to find the longest palindromic substring in a given string by checking every possible substring and verifying if it is a palindrome

## Source Code

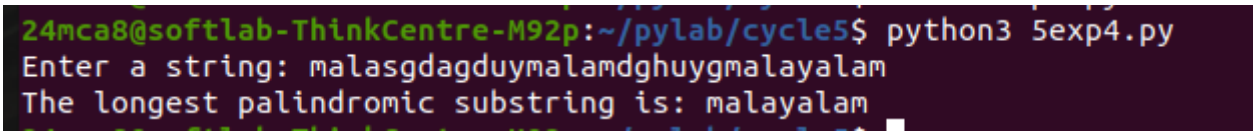
```
import palindrome
def longest_palindromic_substring(s: str) -> str:
    longest = ""

    for i in range(len(s)):
        for j in range(i + 1, len(s) + 1):
            substring = s[i:j]
            if palindrome.is_palindrome(substring) and len(substring) > len(longest):
```

```
            longest = substring

    return longest
input_string = input("Enter a string: ")
result = longest_palindromic_substring(input_string)
print(f"The longest palindromic substring is: {result}")
```

## Output




## Experiment 5

**Aim:**Create a package graphics with modules rectangle, circle and sub-package 3D-graphics 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)

## Source Code

```
import graphics.rectangle as rect
import graphics.circle as circ
from graphics.graphics_3d.cuboid import *
from graphics.graphics_3d.sphere import *

# Rectangle
length = int(input("enter the length of the rectangle: "))
width = int(input("enter the width of the rectangle: "))
print(f"Rectangle Area: {rect.area(length, width)}")
print(f"Rectangle Perimeter: {rect.perimeter(length, width)}")

# Circle
radius = int(input("enter the radius of the circle: "))
print(f"Circle Area: {circ.area(radius)}")
print(f"Circle Perimeter (Circumference): {circ.perimeter(radius)}")

# Cuboid
length = int(input("enter the length of the cuboid: "))
width = int(input("enter the width of the cuboid: "))
height = int(input("enter the height of the cuboid: "))
print(f"Cuboid Volume: {volume(length,width,height)}")

# Sphere
sphere_radius = int(input("enter the radius of the sphere: "))
print(f"Sphere Volume: {volume1(radius)}")
```

**graphics**

**circle.py**

```
import math

def area(radius):
    """Calculate the area of a circle."""
    return math.pi * radius ** 2

def perimeter(radius):
    """Calculate the perimeter (circumference) of a circle."""
    return 2 * math.pi * radius
```

**rectangle.py**

```
def area(length, width):
    """Calculate the area of a rectangle."""
    return length * width

def perimeter(length, width):
    """Calculate the perimeter of a rectangle."""
    return 2 * (length + width)
```

**graphics_3d**

**cuboid.py**

```
def volume(length,width,height):
        return length * width * height
```

**sphere.py**

```
import math
def volume1(radius):
    """Calculate the volume of a sphere."""
    return (4/3) * math.pi * radius ** 3
```

## Output