ANSWERS ASSIGNMENT – 14

TOPICS – DATE & TIME IN PYTHON

1. **A Python Program to measure the time in second since the epoch.**

Solution:

```
import time
seconds = time.time()
print("Seconds since epoch =", seconds)
```

2. **A Python Program to get date and time from the epoch time.**

Solution:

```
import time

# seconds passed since epoch
seconds = 1545925769.9618232
local_time = time.ctime(seconds)
print("Local time:", local_time)
```

3. **A Python Program to convert epoch time into corresponding date and time.**

Solution:

4. **A Python Program to know the current date and time using ctime() function.**

Solution:

```
from time import time, ctime
t = time()
print("epoch Time: ", t)
print(ctime(t))
```

5. **A Python Program to know the local date and time.**

Solution:

```
############
from datetime import date
today = date.today()
print("Today's date:", today)

############
# attributes of now() for timezone

import datetime
now = datetime.datetime.now()
print("Current date and time : ")
print(now.strftime("%Y-%m-%d %H:%M:%S"))

############
# importing time module
import time

# If secs parameter
# is not given then
# the current time as
# returned by time.time() method
# is used

# Convert the current time in seconds
# since the epoch to a
# time.struct_time object in Local time
obj = time.localtime()

# Print the time.struct.time object
print(obj)

# We can change it to
# Day Mon date Hour:Min:Sec year
# format using time.asctime() method
t = time.asctime(obj)
print(t)
```

6. **A Python Program to know today's date and time.**

Solution:

```
# attributes of now() for timezone

import datetime
now = datetime.datetime.now()
print("Current date and time : ")
print(now.strftime("%Y-%m-%d %H:%M:%S"))


import datetime
datetime.datetime.now()

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

7. **A Python Program to datetime object by combining date and time objects.**

Solution:

8. **A Python Program to a datetime object and then change its content.**

Solution:

9. **A Python Program to convert date into a required string format.**

Solution:

10. **A Python Program to find the day of the year and the week day name.**

Solution:

```
import datetime

x = datetime.datetime(2020, 5, 20)
print(x.strftime("%b %d %Y %H:%M:%S"))
```

11. A Python Program to format the time using strftime() method.

Solution:

```
# Program To show How can we use different derivatives
# Multiple at a time and single at a time

# importing the srtftime() and gmtime()
# if not used the gm time, time changes
# to the local time

from time import gmtime, strftime

# using simple format of showing time
s = strftime("%a, %d %b %Y %H:%M:%S", gmtime())
print("Example 1:", s)

print()

# only chnge in this is the full names
# and the representation
s = strftime("%A, %D %B %Y %H:%M:%S", gmtime())
print("Example 2:", s)
print()

# this will show you the preferd date time format
s = strftime("%c")
print("Example 3:", s)
print()

# this will tell about the centuries
s = strftime("%C")
print("Example 4:", s)
print()

# MOTY: month of the year
# DOTY: Day of the year
# Simple representation
# % n - new line
s = strftime("%A, %D %B %Y, %r, %nMOTY:%m %nDOTY:")
print("Example 5:", s)


print()
```

```
# % R - time in 24 hour notation
s = strftime(" %R ")
print("Example 6:", s)

print()

# % H - hour, using a 24-hour clock (00 to 23) in Example 1, 2, 3
# % I - hour, using a 12-hour clock (01 to 12)
s = strftime("%a, %d %b %Y %I:%M:%S + 0000", gmtime())
print("Example 7:", s)

print()

# % T - current time, equal to % H:% M:% S
s = strftime("%r, %T ", gmtime())
print("Example 8:", s)
print()

# % u an % U use (see difference)
s = strftime("%r, %u, %U")
print("Example 9:", s)

print()

# use of % V, % W, % w
s = strftime("%r, %V, %W, %w")
print("Example 10:", s)

print()

# use of % x, % X, % y, % Y
s = strftime("%x, %X, %y, %Y")
print("Example 11:", s)
print()

# use of % Z, % z
s = strftime("%r, %z, %Z")
print("Example 12:", s)
```

12. A Python Program to accept a date from the keyboard and display the day of the week.

Solution:

```
# the week for a given date
import datetime
import calendar


def findDay(date):
    born = datetime.datetime.strptime(date, '%d %m %Y').weekday()
    return (calendar.day_name[born])


# Driver program
date = input("Enter the Date??")
print(findDay(date))
```

13. A Python Program to find the difference in number of days, weeks and months between two given dates.

Solution:

14. A Python Program to find the difference between two dates along with times.

Solution:

```
from datetime import datetime

# create two dates with year, month, day, hour, minute, and second
date1 = datetime(2017, 6, 21, 18, 25, 30)
date2 = datetime(2017, 5, 16, 8, 21, 10)

# Difference between two dates
diff = date1 - date2
print("Difference: ", diff)
```

15. A Python Program to find future date and time from an existing date and time.

Solution:

16. A Python Program to display the next 10 dates continuously.

Solution:

```
import datetime
base = datetime.datetime.today()
for x in range(0, 10):
      print(base + datetime.timedelta(days=x))
```

17. A Python Program to accept date of births of two persons and determining the older person.

Solution:

18. A Python Program to sort a group of given dates in ascending orders.

Solution:

```
# Import the datetime module
from datetime import datetime

dates = ["23 Jun 2018", "02 Dec 2017", "11 Jun 2018",
          "01 Jan 2019", "10 Jul 2016", "01 Jan 2007"]

# Function to print the data stored in the list
def printDates(dates):
    for i in range(len(dates)):
        print(dates[i])


# Sort the list in ascending order of dates
dates.sort(key=lambda date: datetime.strptime(date, '%d %b %Y'))
# Print the dates in a sorted order
printDates(dates)
```

19. A Python Program to generate random numbers in a range with some time delay between each number.

Solution:

```
############ Recheck

import random
import time
num1 = random.randrange(9)

for num in range(num1):
    time.sleep(1)
    print("Random Number: ", num)
```

20. A Python Program to find the execution time of program.

Solution:

```
import time

def ExecutionTime(n):
    starttime = time.time()
    sum = 0
    for i in range(1, n+1):
        sum += i
    endtime = time.time()
    return sum, endtime - starttime
n = 5
print("\nTime to sum of 1 to", n, "and required time to calculate is :"
,ExecutionTime(n))

# Another example:

# importing the modules
from datetime import datetime
import math

N = int(input("Enter the value of N: "))


starttime = datetime.now()
```

```
s = math.factorial(N)

print("factorial of the number:", s)

endtime = datetime.now()
e = endtime - starttime

print("The execution time for factorial program: ", e)
```

21. A Python Program to enter a year number and display whether it is leap or not.

Solution:

```
def checkYear(year):
    # Return true if year is a multiple
    # of 4 and not multiple of 100.
    # OR year is multiple of 400.
    import calendar
    return (calendar.isleap(year))


# Driver Code
year = int(input("Enter the Year??"))
if (checkYear(year)):
    print("Leap Year...")
else:
    print("Not a Leap Year...")
```

22. A Python Program to display the calendar for a given month and year.

Solution:

```
# import module
import calendar

yy = int(input("Please Enter the Year??"))
mm = int(input("Please Enter the Month??"))
# display the calendar
print(calendar.month(yy, mm))
```

23. A Python Program to display the calendar for all months of a given year.

Solution:

```
# program to display calendar of given year

# import module
import calendar

yy = int(input("Please Enter the Year??"))

# display the calendar
print(calendar.calendar(yy))
```