**8**

#### 1: Subtract a week ( 7 days)  from a given date in Python

#### da=int(input("Number of days to be subtracted: "))

#### def subtractDays(d):

#### from datetime import datetime,timedelta

#### today=datetime.now()

#### deletingDays=timedelta(days=d)

#### dateAfterDeleting=(today-deletingDays).strftime("%d-%m-%Y")

#### print(f"date after subtracting {d} days is: {dateAfterDeleting}")

#### subtractDays(da)

**2:** Add week ( 7 days) and 12 hours to a given date

**Given:**

# 2023-03-22 10:00:00

given\_date = datetime(2023, 3, 22, 10, 0, 0)

**Expected output:**

2023-03-29 22:00:00

**from datetime import datetime,timedelta**

**date=datetime(2023, 3, 22, 10, 0, 0)**

**AfterAdding=date+timedelta(days=7,hours=12)**

**print(f"After Adding 7 days and 12 hours date with time: {AfterAdding}")**

**3:** Print ten dates, each two a week apart, starting from today, in the form YYYY-MM-DD.

def addingWeekApart(weeks):

from datetime import datetime,timedelta

t=datetime.now()

l=[7\*i for i in range(1,weeks+1)]

for i in l:

AfterAdding7=t+timedelta(days=i)

print(AfterAdding7.strftime("%Y-%m-%d"))

weeks=int(input("Enter numbers of weeks to be added: "))

addingWeekApart(weeks)

**4: Calculate number of days between two given dates**

**Given:**

# 2020-02-25

date\_1 = datetime(2020, 2, 25)

# 2020-09-17

date\_2 = datetime(2020, 9, 17)

**Expected output:**

205 days

from datetime import datetime,timedelta

date\_1 = datetime(2020, 2, 25)

date\_2 = datetime(2020, 9, 17)

daysDifference=(date\_2-date\_1)

print(daysDifference)

**5:** Write a Python script to display the

a) Current date and time

b) Current year in full

c) Month of year full name

d) Weekday of the week- ex: 5

e) Day of the month

f) Day of week in full name- ex: Friday

from datetime import datetime

todayDate=datetime.now()

print(f"a) Current date and time: {todayDate}")

print(f"b) Current year in full: {todayDate.strftime('%Y')}")

print(f"c) Month of year full name: {todayDate.strftime('%B')}")

print(f"d) Weekday of the week: {todayDate.strftime('%w')}")

print(f"e) Day of the month : {todayDate.strftime('%d')}")

print(f"e) Day of week in full name : {todayDate.strftime('%A')}")

**6:** Python program to convert a string to datetime:

**'Jul 1 2016 2:43AM'** into **2016-07-01 02:43:00**

from datetime import datetime

dateGiven='Jul 1 2016 2:43AM'

dateExtracted=datetime.strptime(dateGiven,'%b %d %Y %I:%M%p')

print(dateExtracted)

**7:** Python program to convert Year/Month/Day to Day of Year.

from datetime import datetime

date='2024/09/14'

dateExtracted=datetime.strptime(date,'%Y/%m/%d')

print(f"Date converted from given {date} : {dateExtracted} ")

days=dateExtracted.strftime("%j")

print("Day of Year: ",days)

**8: Follow the steps:**

Create a class, Triangle. Its \_\_init\_\_() method should take self, angle1, angle2, and angle3 as arguments. Make sure to set these appropriately in the body of the \_\_init\_\_()method.

Create a variable named number\_of\_sides and set it equal to 3.

Create a method named check\_angles. The sum of a triangle's three angles is It should return True if the sum of self.angle1, self.angle2, and self.angle3 is equal 180, and False otherwise.

Create a variable named my\_triangle and set it equal to a new instance of your Triangle class. Pass it three angles that sum to 180 (e.g. 90, 30, 60).

Print out my\_triangle.number\_of\_sides and print out my\_triangle.check\_angles().

class Triangle:

def \_\_init\_\_(self,angle1,angle2,angle3):

self.angle1=angle1

self.angle2=angle2

self.angle3=angle3

print("constructed instantiated.")

number\_of\_sides=3

def check\_angles(self):

if(self.angle1+self.angle2+self.angle3==180):

print("Sum of Angles is 180.")

else:

print("Sum of Angles is not 180.")

def \_\_del\_\_(self):

print("Object is destructed.")

my\_Triangle=Triangle(30,80,70)

print("Number of sides of triangle is ",my\_Triangle.number\_of\_sides)

my\_Triangle.check\_angles()

del my\_Triangle

9: Define a class called Songs, it will show the lyrics of a song. Its \_\_init\_\_() method should have two arguments:self and lyrics.lyrics is a list. Inside your class create a method called sing\_me\_a\_song that prints each element of lyrics on his own line. Define a varible:

happy\_bday = Song(["May god bless you, ",

"Have a sunshine on you,",

"Happy Birthday to you !"])

Call the sing\_me\_song method on this variable.

class Songs:

def \_\_init\_\_(self,lyrics):

self.lyrics=lyrics

def sing\_me\_a\_song(self):

for i in self.lyrics:

print(i)

lyrics1=["May god bless you, ","Have a sunshine on you,","Happy Birthday to you !"]

happy\_bday = Songs(lyrics1)

print(f"given lyrics: {happy\_bday.lyrics}")

print(happy\_bday.sing\_me\_a\_song())

10: Define a class called Lunch.Its \_\_init\_\_() method should have two arguments:selfanf menu.Where menu is a string. Add a method called menu\_price.It will involve a ifstatement:

* if "menu 1" print "Your choice:", menu, "Price 12.00", if "menu 2" print "Your choice:", menu, "Price 13.40", else print "Error in menu".

To check if it works define: Paul=Lunch("menu 1") and call Paul.menu\_price().

class Lunch:

def \_\_init\_\_(self,menu):

self.menu =menu

def menu\_price(self):

if(self.menu=="menu 1"):

print("Your choice:", self.menu, ", Price 12.00")

elif(self.menu=="menu 2"):

print("Your choice:", self.menu, ", Price 13.40")

else:

print("Error in menu")

Paul=Lunch("menu 1")

Paul.menu\_price()

11: Write a Python class which has two methods get\_String and print\_String. get\_String accept a string from the user and print\_String print the string in upper case.

class String\_upper:

def get\_String(self):

string=input("Enter String: ")

self.string=string

def print\_String(self):

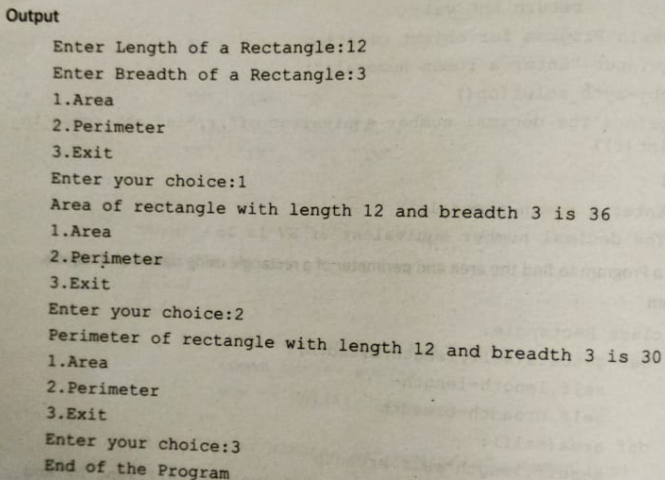
print(f"Upper case of entered string is: {self.string.upper()}")

s1=String\_upper()

s1.get\_String()

s1.print\_String()

12: Write a program to find the area and perimeter of a rectangle using classes and objects. Program output should be like this:



class Rectangle:

def \_\_init\_\_(self):

length=int(input("Enter length of a Rectangle:"))

breadth=int(input("Enter breadth of a Rectangle:"))

c=["1.Area","2.Perimeter","3.exit"]

for item in c:

print(item)

choice=int(input("Enter choice: "))

self.length=length

self.breadth=breadth

self.choice=choice

while True:

if choice==1:

print(f"Area of rectangle with length {self.length} and breadth {self.breadth} is {self.length \* self.breadth}.")

break

elif choice==2:

print(f"Perimeter of rectangle with length {self.length} and breadth {self.breadth} is {2\*(self.length+self.breadth)}.")

break

elif choice==3:

print("Exit of the program.")

break

r1=Rectangle()