

# Programs on Classes and Objects, Classes and Functions

1. Write a Program that prints the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate functions for each operation where the real and imaginary parts are entered by user

```
In [29]: class Complex():
        """This is a class Complex representing the real
        and imaginary parts of a complex number"""

    def read_complex(num):
        num.a=float(input("real part:"))
        num.b=float(input("imaginary part:"))

    def summ(num1,num2):
        summ=Complex() #sum of 2 complex no is an object of class complex
        summ.a=(num1.a+num2.a)
        summ.b=(num1.b+num2.b)
        return(summ)

    def diff(num1,num2):
        difff=Complex() #difference of 2 complex no is an object of class complex
        difff.a=(num1.a-num2.a)
        difff.b=(num1.b-num2.b)
        return(difff)

    def product(num1,num2):
        pro=Complex() #product of 2 complex no is an object of class complex
        pro.a=(num1.a*num2.a)-(num1.b*num2.b)
        pro.b=(num1.a*num2.b)+(num1.b*num2.a)
        return(pro)

    def print_complex(num):
        if(num.b>0):
            print("%g+%gi"%(num.a,num.b))
        elif(num.b<0):
            print("%g%gi"%(num.a,num.b))
        else:
            print("%g"%(num.a))

    num1=Complex() #num is an object of class Complex
    print("Enter the first complex no:")
    read_complex(num1)
    num2=Complex()
    print("Enter the second complex no:")
    read_complex(num2)

    sum_res=summ(num1,num2);
    print("Sum=",end="")
    print_complex(sum_res)

    diff_res=diff(num1,num2);
    print("Difference=",end="")
    print_complex(diff_res)

    pro_res=product(num1,num2);
    print("Product=",end="")
    print_complex(pro_res)
```

```
Enter the first complex no:  
real part:2  
imaginary part:3  
Enter the second complex no:  
real part:-6  
imaginary part:-5  
Sum=-4-2i  
Difference=8+8i  
Product=3-28i
```

2. Write a Program that defines a class Point that represents a mathematical point, Write Python function to Find the distance between two point objects.

```

In [1]: import math

class Point:
    """ This is a class Point representing a coordinate point """

    #function to read the attributes of the Point object
    def read_point(p):
        p.x=float(input("x coordinate:"))
        p.y=float(input("y coordinate:"))

    #function to print the attributes of the Point object
    def print_point(p):
        print("(%g,%g)"%(p.x,p.y))

    #function which takes two objects of Point class as arguments
    #and computes the Euclidean distance between them
    def distance(p1,p2):
        d=math.sqrt((p1.x-p2.x)**2+(p1.y-p2.y)**2)
        return d

p1=Point() #create first object print("Enter First point:")
print("Enter the first point:")
read_point(p1) #read x and y for p1

p2=Point() #create second object print("Enter Second point:")
print("Enter the second point:")
read_point(p2) #read x and y for p2

dist=distance(p1,p2) #compute distance

print("First point is:")
print_point(p1) #print p1

print("Second point is:")
print_point(p2) #print p2

print("Distance is: %g"%dist) #print the Euclidean distance between p1 and p2

```

Enter the first point:  
x coordinate:3  
y coordinate:4  
Enter the second point:  
x coordinate:5  
y coordinate:-6  
First point is:  
(3,4)  
Second point is:  
(5,-6)  
Distance is: 10.198

3. Write a Program that defines a class Point that represents a mathematical point, Write Python function to Find Whether it Lies in the First, Second, Third or Fourth Quadrant of x - y Plane

```

In [3]: import math

class Point:
    """ This is a class Point representing a coordinate point """

    #function to read the attributes of the Point object
    def read_point(p):
        p.x=float(input("x coordinate:"))
        p.y=float(input("y coordinate:"))

    #function to print the attributes of the Point object
    def print_point(p):
        print("(%g,%g)"%(p.x,p.y))

    #function which takes an object of Point class as arguments
    #and determines to which quadrant it lies
    def quadrant(p1):
        if p1.x>0 and p1.y>0:
            print("Point lies in I Quadrant")
        elif p1.x<0 and p1.y>0:
            print("Point lies in II Quadrant")
        elif p1.x<0 and p1.y<0:
            print("Point lies in III Quadrant")
        elif p1.x>0 and p1.y<0:
            print("Point lies in IV Quadrant")
        else:
            print("Point is an origin")

p=Point() #create first object
print("Enter the Point:")
read_point(p) #read x and y for p

print("First point is:")
print_point(p) #print p

quadrant(p) #checks for quadrant for p

```

```

Enter the Point:
x coordinate:3
y coordinate:4
First point is:
(3,4)
Point lies in I Quadrant

```

4. Write Python Program to Calculate the Arc Length of an Angle by Assigning Values to the Radius and Angle Data Attributes of the class ArcLength

```
In [6]: import math

class ArcLength():
    """This is a class ArcLength representing the attributes angle and radius"""

    def read(al1):
        al1.radius=float(input("Radius:"))
        al1.angle=float(input("Angle:"))

    def calculate_arc_length(al1):
        if al1.angle >= 360:
            print("Angle is not possible")
        else:
            result =(2* math.pi *al1.radius) *(al1.angle/360)
            return result

al = ArcLength()
print("Enter radius and angle:")
read(al)
res=calculate_arc_length(al)
print("Arc Length is",res)
```

```
Enter radius and angle:
Radius:4.5
Angle:45
Arc Length is 3.5342917352885173
```

5. Write a class Rectangle containing numeric attributes width and height. This class should contain another attribute corner which is an instance of another class Point. Implement following functions –

- A function to print corner point as an ordered-pair
- A function find\_center() to compute center point of the rectangle
- A function resize() to modify the size of rectangle

```

In [8]: class Point:
        """ This is a class Point representing coordinate point"""

class Rectangle:
    """ This is a class Rectangle. Attributes: width, height and Corner Point
        """

def find_center(rect):
    """Returns a Point at the center of a Rectangle.
    rect: Rectangle object
    returns: new Point object"""
    center=Point()
    center.x = rect.corner.x + rect.width/2
    center.y = rect.corner.y + rect.height/2
    return center

def resize(rect, w, h):
    """Modifies the Rectangle by adding to its width and height.
    rect: Rectangle object.
    dwidth: change in width (can be negative).
    dheight: change in height (can be negative). """
    rect.width +=w
    rect.height+=h

def print_point(p):
    print("(%g,%g)"%(p.x, p.y))

box=Rectangle() #create Rectangle object
box.corner=Point() #define an attribute corner for box
box.width=100 #set attribute width to box
box.height=200 #set attribute height to box
box.corner.x=0 #corner itself has two attributes x and y
box.corner.y=0 #initialize x and y to 0

print("Original Rectangle is:")
print("width=%g, height=%g"%(box.width, box.height))

center=find_center(box)
print("The center of rectangle is:")
print_point(center)
resize(box,50,70)
print("Rectangle after resize:")
print("width=%g, height=%g"%(box.width, box.height))

center=find_center(box)
print("The center of resized rectangle is:")
print_point(center)

```

```

Original Rectangle is:
width=100, height=200
The center of rectangle is:
(50,100)
Rectangle after resize:
width=150, height=270
The center of resized rectangle is:
(75,135)

```

6. Write a class `Rectangle` containing numeric attributes `width` and `height`. This class should contain another attribute `corner` which is an instance of another class `Point`. Write a function named `move_rectangle` that takes a `Rectangle` and two numbers named `dx` and `dy`. It should change the location of the rectangle by adding `dx` to the `x` coordinate of `corner` and adding `dy` to the `y` coordinate of `corner`.

```
In [9]: class Point:
        """ This is a class Point representing coordinate point """

        class Rectangle:
            """ This is a class Rectangle. Attributes: width, height and Corner Point """

        def print_point(p):
            print("(%g,%g"%(p.x, p.y))

        def move_rectangle(box,dx,dy):
            box.corner.x+=dx
            box.corner.y+=dy

        box=Rectangle() #an object for class Rectangle
        box.width=100
        box.height=150
        box.corner=Point() # an attribute corner for box

        box.corner.x=int(input("x-coordinate:"))
        box.corner.y=int(input("y-coordinate:"))

        print("The original corner is:")
        print_point(box.corner)

        dx=int(input("dx:"))
        dy=int(input("dy:"))

        move_rectangle(box,dx,dy);

        print("After changing the corner is:")
        print_point(box.corner)

x-coordinate:3
y-coordinate:4
The original corner is:
(3,4)
dx:6
dy:-7
After changing the corner is:
(9,-3)
```

7. Write a definition for a class named `Circle` with attributes `center` and `radius`, where `center` is a `Point` object and `radius` is a number. Instantiate a `Circle` object that represents a circle with its center at (150, 100) and radius 75. Write a function named `point_in_circle` that takes a `Circle` and a `Point` and returns `True` if the `Point` lies in or on the boundary of the circle



```

In [15]: import math

class Point:
    """This is a class Point Representing a point with
    Attributes:x and y"""

def read_point(p):
    p.x=float(input("enter the x-coordinate:"))
    p.y=float(input("enter the y-coordinate:"))

class Circle:
    """This is a class Circle Representing a circle with
    Attributes:radius and center x and y"""

def distance(p1,p2):
    d=math.sqrt((p1.x-p2.x)**2+(p1.y-p2.y)**2)
    return d

def point_in_circle(c,c1):
    """returns true if point lies in the
    boundary of the circle"""

    d=distance(c.center,c1)
    print("distance:",d)
    if(d<=c.radius):
        return True
    else:
        return False

c=Circle()          #c is an instace/object of class Circle
c.center=Point()    #center is an attribute of object c

c.center.x=150
c.center.y=100
c.radius=75

c1=Point()

print("Enter the point:")
read_point(c1)

if(point_in_circle(c,c1)):
    print("Point lies inside the circle.")
else:
    print("Point does not lies inside the circle.")

```

```

Enter the point:
enter the x-coordinate:151
enter the y-coordinate:101
distance: 1.4142135623730951
Point lies inside the circle.

```

8. Write a Program that creates a class called Time with attributes hour, minute and second. Write the following functions

- A function to read the attributes
- A function to print the time in format
- A function to add 2 time objects

```

In [19]: class Time():
          """Represents the time of day
          attributes:hour,min,sec"""

          #function to read time
          def read_time(time):
              time.hour,time.minute,time.second=input("Enter the time-").split(":")
              time.hour,time.minute,time.second=int(time.hour),int(time.minute),int(time
              .second)

          #function to print time
          def print_time(time):
              print("%02d:%02d:%02d"%(time.hour,time.minute,time.second))

          #converts time to integer and returns the total time in sec
          def time_to_int(time):
              minutes = time.hour * 60 + time.minute
              seconds = minutes * 60 + time.second
              return seconds

          #converts total seconds to time with 24h format
          def int_to_time(seconds):
              time = Time()
              minutes, time.second = divmod(seconds, 60)
              time.hour, time.minute = divmod(minutes, 60)
              return time

          #adds the two time
          def add_time(t1,t2):
              seconds = time_to_int(t1) + time_to_int(t2)
              return int_to_time(seconds)

          #to add two time
          print("To add 2 time's:")
          time1,time2=Time(),Time()
          print("Enter the first time in format hh:mm:ss:")
          read_time(time1)
          print("Enter the second time in format hh:mm:ss:")
          read_time(time2)

          total_time=add_time(time1,time2)
          print("Total time=",end="")
          print_time(total_time)

```

```

To add 2 time's:
Enter the first time in format hh:mm:ss:
Enter the time-1:34:56
Enter the second time in format hh:mm:ss:
Enter the time-3:56:56
Total time=05:31:52

```

9. Write a Program that creates a class called Time with attributes hour, minute and second. Write the following functions

- A function to read the attributes
- A function to print the time in format
- A function to sub 2 time objects where the first object represents start of the event and second object represents end of the event and finds the duration between them

```

In [21]: class Time():
          """Represents the time of day
          attributes:hour,min,sec"""

          #function to read time
          def read_time(time):
              time.hour,time.minute,time.second=input("Enter the time-").split(":")
              time.hour,time.minute,time.second=int(time.hour),int(time.minute),int(time
              .second)

          #function to print time
          def print_time(time):
              print("%02d:%02d:%02d"%(time.hour,time.minute,time.second))

          #converts time to integer and returns the total time in sec
          def time_to_int(time):
              minutes = time.hour * 60 + time.minute
              seconds = minutes * 60 + time.second
              return seconds

          #converts total seconds to time with 24h format
          def int_to_time(seconds):
              time = Time()
              minutes, time.second = divmod(seconds, 60)
              time.hour, time.minute = divmod(minutes, 60)
              return time

          #adds the two time
          def sub_time(t2,t1):
              seconds = time_to_int(t2) - time_to_int(t1)
              return int_to_time(seconds)

          #to add two time
          print("To subtract 2 time's:")
          time1,time2=Time(),Time()
          print("Enter the start time in format hh:mm:ss:")
          read_time(time1)
          print("Enter the end time in format hh:mm:ss:")
          read_time(time2)

          duration_time=sub_time(time2,time1)
          print("Duration time=",end="")
          print_time(duration_time)

```

```

To subtract 2 time's:
Enter the start time in format hh:mm:ss:
Enter the time-2:30:00
Enter the end time in format hh:mm:ss:
Enter the time-4:45:00
Duration time=02:15:00

```

10. Write a Program that creates a class called Time with attributes hour, minute and second. Write the following functions

- A function to read the attributes
- A function to print the time in format
- Write a function called mul\_time that takes a Time object and a number and returns a new Time object that contains the product of the original Time and the number.

```

In [26]: import copy

class Time():
    """Represents the time of day
    attributes:hour,min,sec"""

    #function to read time
    def read_time(time):
        time.hour,time.minute,time.second=input("Enter the time-").split(":")
        time.hour,time.minute,time.second=int(time.hour),int(time.minute),int(time
        .second)

    #function to print time
    def print_time(time):
        print("%02d:%02d:%02d"%(time.hour,time.minute,time.second))

    #converts time to integer and returns the total time in sec
    def time_to_int(time):
        minutes = time.hour * 60 + time.minute
        seconds = minutes * 60 + time.second
        return seconds

    #converts total seconds to time with 24h format
    def int_to_time(seconds):
        time = Time()
        minutes, time.second = divmod(seconds, 60)
        time.hour, time.minute = divmod(minutes, 60)
        return time

    # pure function mul_time():
    def mul_time(time,num):
        time_int = time_to_int(time) * num
        new_time = int_to_time(time_int)
        if new_time.hour > 24:
            new_time.hour = new_time.hour % 24
        return new_time

    #to multiply time by a number
    print("\nTo multiply time by a give no.")
    time=Time()
    print("Enter the time in format hh:mm:ss:")
    read_time(time)
    secs=int(input("Enter the additonal seconds:"))
    res=mul_time(time,secs)
    print("New time after multiplication:",end="")
    print_time(res)

```

```

To multiply time by a give no.
Enter the time in format hh:mm:ss:
Enter the time-2:30:45
Enter the additonal seconds:45
New time after multiplication:17:03:45

```

11. Write a Program that creates a class called Time with attributes hour, minute and second. Write the following functions

- A function to read the attributes
- A function to print the time in format
- Write a boolean function called is\_after that takes two Time objects, t1 and t2, and returns True if t1 follows t2 chronologically and False otherwise.



```

In [28]: class Time():
        """Represents the time of day
        attributes:hour,min,sec"""

        #function to read time
        def read_time(time):
            time.hour,time.minute,time.second=input("Enter the time-").split(":")
            time.hour,time.minute,time.second=int(time.hour),int(time.minute),int(time
            .second)

        #function to print time
        def print_time(time):
            print("%02d:%02d:%02d"%(time.hour,time.minute,time.second))

        #converts time to integer and returns the total time in sec
        def time_to_int(time):
            minutes = time.hour * 60 + time.minute
            seconds = minutes * 60 + time.second
            return seconds

        #converts total seconds to time with 24h format
        def int_to_time(seconds):
            time = Time()
            minutes, time.second = divmod(seconds, 60)
            time.hour, time.minute = divmod(minutes, 60)
            return time

        def is_after(t1,t2):
            a=time_to_int(t1)
            b=time_to_int(t2)
            return a>b

        time1,time2=Time(),Time()
        print("Enter the start time in format hh:mm:ss:")
        read_time(time1)
        print("Enter the end time in format hh:mm:ss:")
        read_time(time2)
        if(is_after(time1,time2)):
            print("Time1 follows Time2")
        else:
            print("Time2 follows Time1")

```

```

Enter the start time in format hh:mm:ss:
Enter the time-12:30:45
Enter the end time in format hh:mm:ss:
Enter the time-11:30:45
Time1 follows Time2

```

12. Write a Program that creates a class called Time with attributes hour, minute and second. Write the following functions

- A function to read the attributes
- A function to print the time in format
- A modifier increment, which adds a given number of seconds to a Time object

```
In [32]: class Time():
        """Represents the time of day
        attributes: hour, min, sec"""

        #function to read time
        def read_time(time):
            time.hour, time.minute, time.second = input("Enter the time-").split(":")
            time.hour, time.minute, time.second = int(time.hour), int(time.minute), int(time
            .second)

        #function to print time
        def print_time(time):
            print("%02d:%02d:%02d"%(time.hour, time.minute, time.second))

        #modifier that increments
        def increment(t, seconds):
            t.second += seconds

            while t.second >= 60:
                t.second -= 60
                t.minute += 1

            while t.minute >= 60:
                t.minute -= 60
                t.hour += 1

        time1 = Time()
        print("Enter the time in format hh:mm:ss:")
        read_time(time1)
        sec = int(input("Enter the additional seconds::"))
        increment(time1, sec)
        print_time(time1)
```

```
Enter the time in format hh:mm:ss:
Enter the time-12:30:45
Enter the additional seconds::120
12:32:45
```

13. Write a Program that creates a class called Time with attributes hour, minute and second. Write the following functions

- A function to read the attributes
- A function to print the time in format
- Write a “pure” version of increment that creates and returns a new Time object rather than modifying the parameter.

```
In [33]: import copy

class Time():
    """Represents the time of day
    attributes: hour, min, sec"""

    #function to read time
    def read_time(time):
        time.hour, time.minute, time.second = input("Enter the time-").split(":")
        time.hour, time.minute, time.second = int(time.hour), int(time.minute), int(time
        .second)

    #function to print time
    def print_time(time):
        print("%02d:%02d:%02d"%(time.hour, time.minute, time.second))

    #pure function that increments
    def increment(t1, seconds):
        t = copy.copy(t1)
        t.second += seconds

        while t.second >= 60:
            t.second -= 60
            t.minute += 1

        while t.minute >= 60:
            t.minute -= 60
            t.hour += 1

        return t

time1 = Time()
print("Enter the time in format hh:mm:ss:")
read_time(time1)
sec = int(input("Enter the additional seconds::"))
res_time = increment(time1, sec)
print_time(res_time)
```

```
Enter the time in format hh:mm:ss:
Enter the time-12:30:45
Enter the additional seconds::129
12:32:54
```