**Task Seven**

**Classes & Objects**

1) Write a program that calculates and prints the value according to the given formula:

Q= Square root of [(2\*C\*D)/H]. Following are the fixed values of C and H: C is 50. H is 30. D is a variable whose values should be input to your program in a comma-separated sequence.

D = list(map(int, input("Enter multiple values\n").split(', ')))

for i in D:

    Q=((2\*50\*i)/30)\*\*0.5

    print(Q)

2) Define a class named Shape and its subclass Square. The Square class has an init function which takes length as argument. Both classes have an area function which can print the area of the shape where Shape’s area is 0 by default.

**class** Shape:

**def** \_\_init\_\_(self):

        pass

**def** area(self):

        return 0

**class** Square(Shape):

**def** \_\_init\_\_(self, length):

        self.length=length

**def** area(self):

        print("Area is ",self.length\*self.length)

q=Shape()

d=Square(2)

d.area()

3) Create a class to find three elements that sum to zero from a set of n real numbers.

**class** ThreeNos:

**def** summ(self,arr):

        arr, reslt, i = sorted(arr), [], 0

        while i < len(arr) - 2:

            j, k = i + 1, len(arr) - 1

            while j < k:

                if arr[i] + arr[j] + arr[k] < 0:

                    j += 1

                elif arr[i] + arr[j] + arr[k] > 0:

                    k -= 1

                else:

                    reslt.append([arr[i], arr[j], arr[k]])

                    j, k = j + 1, k - 1

                    while j < k and arr[j] == arr[j - 1]:

                         j += 1

                    while j < k and arr[k] == arr[k + 1]:

                        k -= 1

            i += 1

            while i < len(arr) - 2 and arr[i] == arr[i - 1]:

                i += 1

        return reslt

c=ThreeNos()

print(c.summ([-25,-10,-7,-3,2,4,8,10]))

4) Create a Time class and initialize it with hours and minutes. Create a method addTime which should take two Time objects and add them. Create another method displayTime which should print the time. Also create a method displayMinute which should display the total minutes in the Time.

**class** Time:

**def** \_\_init\_\_(self,hour,min):

        self.hour=hour

        self.min=min

**def** addtime(t1,t2):

        t3 = Time(0,0)

        t3.hour=t1.hour+t2.hour

        t3.min=t1.min+t2.min

        while t3.min>= 60:

            t3.hour+= 1

            t3.min-= 60

        return t3

**def** displaytime(self):

        print("Total time is {} hours and {} minutes".format(self.hour,self.min))

**def** displayminute(self):

        print((self.hour\*60)+self.min," minutes")

c1=Time(1,2)

c2=Time(2,5)

c=Time.addtime(c1,c2)

c.displaytime()

c.displayminute()

5) Write a Person class with an instance variable “age” and a constructor that takes an integer as a parameter. The constructor must assign the integer value to the age variable after confirming the argument passed is not negative; if a negative argument is passed then the constructor should set age to 0 and print “Age is not valid, setting age to 0”. In addition, you must write the following instance methods:

* yearPasses() should increase age by the integer value that you are passing inside the function.
* amIOld() should perform the following conditional actions: If age is between 0 and <13, print “You are young”. If age is >=13 and <=19 , print “You are a teenager”. Otherwise, print “You are old”.

**class** Person:

**def** \_\_init\_\_(self,age):

        self.age=age

**def** checkage(self):

        if self.age<0:

            print("Invalid age")

            return 0

        else:

            print("Age is ",self.age)

**def** yearPasses(self,random):

        self.random=random

        r=self.random+self.age

        print("Age in yearPasses():",r)

**def** amIOld(self):

        if 0<self.age<13:

            print("You are a youngling!")

        elif 13<=self.age<=19:

            print("You are a teen!")

        else:

            print("Oldie!")

c=Person(4)

c.checkage()

c.yearPasses(4)

c.amIOld()