**TASK SEVEN:**

**CLASSES AND 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 the variable whose values should be input to your program in a comma-separated sequence.

import math

C= 50

H = 30

Ds = []

result =[]

D=input("Enter numbers seprates with comma:")

Ds=D.split(",")

Ds = [int(i) for i in Ds]

i=0

l = len(Ds)

while(i<l):

Q = round(math.sqrt((2\*C\*Ds[i])/H))

result. append(Q)

i+=1

print("The values of Q is: ",result)

Output:



2. Define a class named Shape and its subclass Square. The Square class has an init function which takes a 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(object):

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

pass

def area(self):

return 0

class Square(Shape):

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

Shape.\_\_init\_\_(self)

self.length = l

def area(self):

return self.length\*self.length

aSquare= Square(4)

print (aSquare.area())

Output:



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

Input array: [-25,-10,-7,-3,2,4,8,10]

Output: [[-10,2,8],[-7,-3,10]]

class find\_triplets:

def triplets(self, nums):

nums, result, i = sorted(nums), [], 0

while i < len(nums) - 2:

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

while j < k:

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

j += 1

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

k -= 1

else:

result.append([nums[i], nums[j], nums[k]])

j, k = j + 1, k - 1

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

j += 1

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

k -= 1

i += 1

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

i += 1

return result

print(find\_triplets().triplets([-25, -10, -7, -3, 2, 4, 8, 10]))

Output:



4. Create a Time class and initialize it with hours and minutes.

Make a method addTime which should take two time object and add them. E.g.- (2 hour and 50 min)+(1 hr and 20 min) is (4 hr and 10 min)

Make a method displayTime which should print the time.

Make a method DisplayMinute which should display the total minutes in the Time. E.g.- (1 hr 2 min) should display 62 minute.

class Time():

def \_\_init\_\_(self, hours, mins):

self.hours = hours

self.mins = mins

def addTime(t1, t2):

t3 = Time(0,0)

if t1.mins+t2.mins > 60:

t3.hours = (t1.mins+t2.mins)/60

t3.hours = t3.hours+t1.hours+t2.hours

t3.mins = (t1.mins+t2.mins)-(((t1.mins+t2.mins)/60)\*60)

return t3

def displayTime(self):

print ("Time is",self.hours,"hours and",self.mins,"minutes.")

def displayMinute(self):

print ("Displaying in minutes for give time in hours is:",(self.hours\*60)+self.mins)

a = Time(2,50)

b = Time(1,20)

c = Time.addTime(a,b)

c.displayTime()

c.displayMinute()

Output:



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

1. yearPasses() should increase the instance variable by .
2. amIOld() should perform the following conditional actions:
   * If , print You are young..
   * If and , print You are a teenager..
   * Otherwise, print You are old..

Sample Input:

-1

4

10

16

18

Sample Output:

class Person:

age = 0

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

# Add some more code to run some checks on initialAge

if initialAge < 0:

print ("Age is not valid, setting age to 0.")

else:

self.age = initialAge

def amIOld(self):

# Do some computations in here and print out the correct statement to the console

if self.age < 13:

print ("You are young.")

elif self.age >= 13 and self.age < 18:

print ("You are a teenager.")

else:

print ("You are old.")

def yearPasses(self):

# Increment the age of the person in here

self.age += 1

t = int(input())

for i in range(0, t):

age = int(input())

p = Person(age)

p.amIOld()

for j in range(0, 3):

p.yearPasses()

p.amIOld()

print("")

Output:

-1

Age is not valid, setting age to 0.

4