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

import math

C = 50

H = 30

def calculate\_q(D):

result = []

values = D.split(',')

for value in values:

value = int(value)

Q = math.sqrt((2 \* C \* value) / H)

result.append(str(round(Q)))

return result

# Take input for D

D = input("Enter values of D (comma-separated): ")

# Calculate and print Q values

Q\_values = calculate\_q(D)

print("Q values:", ', '.join(Q\_values))

**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):

super().\_\_init\_\_()

self.length = length

def area(self):

return self.length \*\* 2

shape = Shape()

print("Area of Shape:", shape.area())

square = Square(5)

print("Area of Square:", square.area())

**3.** Create a class to find three elements that sum to zero from a set of n real numbers **Input array:** [-25,-10,-7,-3,2,4,8,10]  
**Expected output:** [[-10,2,8],[-7,-3,10]]

class ThreeSum:

def find\_three\_sum(self, nums):

result = []

nums.sort()

for i in range(len(nums) - 2):

if i > 0 and nums[i] == nums[i - 1]:

continue # Skip duplicates

left = i + 1

right = len(nums) - 1

while left < right:

curr\_sum = nums[i] + nums[left] + nums[right]

if curr\_sum < 0:

left += 1

elif curr\_sum > 0:

right -= 1

else:

result.append([nums[i], nums[left], nums[right]])

while left < right and nums[left] == nums[left + 1]:

left += 1 # Skip duplicates

while left < right and nums[right] == nums[right - 1]:

right -= 1 # Skip duplicates

left += 1

right -= 1

return result

# Create an instance of ThreeSum class

three\_sum = ThreeSum()

# Test the find\_three\_sum() method

input\_array = [-25, -10, -7, -3, 2, 4, 8, 10]

output = three\_sum.find\_three\_sum(input\_array)

print("Output:", output)

**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.  
E.g.- (2 hour and 50 min)+(1 hr and 20 min) is (4 hr and 10 min)  
Create another method **displayTime** which should print the time.  
Also create 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, minutes):

self.hours = hours

self.minutes = minutes

def addTime(self, time2):

total\_minutes = self.hours \* 60 + self.minutes + time2.hours \* 60 + time2.minutes

hours = total\_minutes // 60

minutes = total\_minutes % 60

return Time(hours, minutes)

def displayTime(self):

print(f"{self.hours} hr {self.minutes} min")

def displayMinute(self):

total\_minutes = self.hours \* 60 + self.minutes

print(f"Total minutes: {total\_minutes} min")

# Create two Time objects

time1 = Time(2, 50)

time2 = Time(1, 20)

# Add the two Time objects

result = time1.addTime(time2)

# Display the result

result.displayTime()

# Display the total minutes

result.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:I  
f 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**”.

**Sample Input** for **amIOld():** -1  
4  
10

16 18 64 38

**Expected Output for amIOld():**

Age is not valid, setting age to 0. You are young.  
You are young.  
You are a teenager.

You are a teenager. You are old.  
You are old.

Consider the age variable to be set to 38 then: **Sample Input for yearPasses():** 4  
**Expected Output for yearPasses():** 42