**ASSIGNMENT 8**

**Question 1.**

class String:

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

self.strval=""

def getString(self,s):

self.strval=s

def printString(self):

print(self.strval)

str1= String()

s=input("Enter String:")

str1.getString(s)

str1.printString()

**Question 2.**

class Subsets:

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

self.A=A

self.B=[]

def getSubset(self):

for i in range(len(self.A)):

for j in range(i,len(self.A)):

self.B.append(self.A[i:j+1])

def showSubset(self):

print(self.B)

ss1=Subsets([4,5,6])

ss1.getSubset()

ss1.showSubset()

**Question 3.**

class Circle:

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

self.r=r

def findArea(self):

self.Area=3.14\*self.r\*self.r

def showArea(self):

print(self.Area)

C=Circle(10)

C.findArea()

C.showArea()

**Question 4.**

class Rectangle:

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

self.l=l

self.r=r

def findArea(self):

self.Area=self.l\*self.r

def showArea(self):

print(self.Area)

C=Rectangle(10,20)

C.findArea()

C.showArea()

**Question 5.**

class Gen:

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

self.n=n

def genclass(self):

for i in range(self.n):

if(i%7==0):

yield (i)

G= Gen(18)

for i in G.genclass():

print(i)

**Question 6.**

class Shape():

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

self.Area=0

self.r=0

def area(self):

print(self.Area)

class Square(Shape):

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

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

self.l=l

def area(self):

self.Area=self.l\*\*2

print(self.Area)

S=Square(5)

S.area()

**Question 7.**

class Person:

def \_\_init\_\_(self ,n,a):

self.name=n

self.age=a

class Address:

def \_\_init\_\_(self,add,pin):

self.add=add

self.pin=pin

class Contact(Person,Address):

def \_\_init\_\_(self,n,a,add,pin,m):

Person.\_\_init\_\_(self,n,a)

Address.\_\_init\_\_(self,add,pin)

self.m=m

def show(self):

print(self.name,self.age,self.add,self.pin,self.m)

C=Contact("ABC",18,"XYZ",455455,9456564456)

C.show()

**Question 8.**

class py\_solution:

def int\_to\_Roman(self, num):

val = [

1000, 900, 500, 400,

100, 90, 50, 40,

10, 9, 5, 4,

1

]

syb = [

"M", "CM", "D", "CD",

"C", "XC", "L", "XL",

"X", "IX", "V", "IV",

"I"

]

roman\_num = ''

i = 0

while num > 0:

for \_ in range(num // val[i]):

roman\_num += syb[i]

num -= val[i]

i += 1

return roman\_num

print(py\_solution().int\_to\_Roman(1))

print(py\_solution().int\_to\_Roman(4000))

**Question 9.**

class py\_solution:

def roman\_to\_int(self, s):

rom\_val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}

int\_val = 0

for i in range(len(s)):

if i > 0 and rom\_val[s[i]] > rom\_val[s[i - 1]]:

int\_val += rom\_val[s[i]] - 2 \* rom\_val[s[i - 1]]

else:

int\_val += rom\_val[s[i]]

return int\_val

print(py\_solution().roman\_to\_int('MMMCMLXXXVI'))

print(py\_solution().roman\_to\_int('MMMM'))

print(py\_solution().roman\_to\_int('C'))