PYTHON ASSIGNMENT: 8

1. Define a class which has at least two methods: getString: to get a string from console input printString: to print the string in upper case.

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()

2. Write a Python class to get all possible unique subsets from a set of distinct integers. Input : [4, 5, 6] Output : [[], [6], [5], [5, 6], [4], [4, 6], [4, 5], [4, 5, 6]]

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()

3. Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area.

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()

4. Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.

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()

5. Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

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)

6. 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 a area function which can print the area of the shape where Shape's area is 0 by default.

class Shape():

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

self.Area=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()

7. Derive class Contact from the base classes Person and Address and use their methods to print out the contact information.

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()

8. Write a Python class to convert an integer to a roman numeral.

class IntoRoman:

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

self.number=n

def printRoman(self):

num = [1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500, 900, 1000]

sym = ["I", "IV", "V", "IX", "X", "XL", "L", "XC", "C", "CD", "D", "CM", "M"]

i = 12

while (self.number):

div = self.number // num[i]

self.number = self.number%num[i]

while (div):

print(sym[i], end = "")

div -= 1

i -= 1

IR=IntoRoman(252)

IR.printRoman()

9. Write a Python class to convert a roman numeral to an integer.

class RomanintoInt:

def rti(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]]

print (int\_val)

A=RomanintoInt()

A.rti('CD')