Basic python

1. Write a program to print your name.

print ("MANSI")

2. Write a program for a Single line comment and multi-line comments.

# Read the input from the user

# Perform some operations on the input data

# Output the result

print("Multi-line commenting")

3. Define variables for different Data Types int, Boolean, char, float, double and print on the Console.

a = -5

print("Type of a: ", type(a))

b = False

print("Type of b: ", type(b))

c = 5.0

print("Type of c: ", type(c))

String = 'Hello'

print("Type of String: ", type(String))

4. Define the local and Global variables with the same name and print both variables and understand the scope of the variables.

a = 5

# Uses global because there is no local 'a'

def f():

print('Inside f() : ', a)

5. Variable 'a' is redefined as a local

def g():

a = 2

print('Inside g() : ', a)

# Uses global keyword to modify global 'a'

def h():

global a

a = 4 #Value of 'a' modified

print('Inside h() : ', a)

# Global scope

print('global : ', a)

f()

print('global : ', a)

g()

print('global : ', a)

h()

print('global : ', a)

PYTHON OPERATORS

1. Write a function for arithmetic operators(+,-,\*,/)

num1 = input('Enter first number: ')

num2 = input('Enter second number: ')

sum = float(num1) + float(num2)

min = float(num1) - float(num2)

mul = float(num1) \* float(num2)

div = float(num1) / float(num2)

print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))

print('The subtraction of {0} and {1} is {2}'.format(num1, num2, min))

print('The multiplication of {0} and {1} is {2}'.format(num1, num2, mul))

print('The division of {0} and {1} is {2}'.format(num1, num2, div))

1. Write a method for increment and decrement operators(++, --)

a = 0

a += 1

a = a+1

print('The value of a is ',a)

print("INCREMENTED FOR LOOP")

for i in range(0, 5):

print(i)

print("\nDECREMENTED FOR LOOP")

for i in range(4, -1, -1):

print(i)

3. Write a program to find the two numbers equal or not.

a = input('Enter first number: ')

b = input('Enter second number: ')

if a==b:

print("Both numbers are equal")

else:

print("Both numbers are not equal")

1. Program for relational operators (<,<==, >, >==)

a = 9

b = 10

print(a < b) #This operator(<).

print(a <= b)

print(a > b)

print(a >= b)

print(a == b)

print(a != b)

a = float(input('Enter first number: '))

b = float(input('Enter second number: '))

if a > b:

print(a, "is greater ")

else:

print(b, " is greater ")

if a < b:

print(a, "is smaller2 ")

else:

print(b, " is smaller ")

PYTHON LOOPS

1. Write a program to print “Bright IT Career” ten times using for loop.

a = "Bright IT Career"

for i in range(10):

print(a)

2. Write a program to print 1 to 20 numbers using the while loop.

i = 1

while(i<=20):

print(i)

i+= 1

3. Program to equal operator and not equal operators.

a = 5

b = 10

print(a ==b)

print(a != b)

4. Write a program to print the odd and even numbers.

N = [1,2,3,4,5,6,7,8,9,10]

print("Even Numbers: ")

for i in Numbers:

if i % 2 == 0:

print(i, end =" ")

print("\nOdd Numbers:")

for i in N:

if i % 2 != 0:

print(i, end =" ")

print()

5. Write a program to print largest number among three numbers.

k = 40

a = 50

s = 90

if k >= a and k >= s:

largest = k

elif a >= k and a >= s:

largest = a

else:

largest = s

print("Largest number is: ",largest)

6. Write a program to print even number between 10 and 20 using while.

a = 10

b = 20

print("Even Numbers Between 10 to 20: ",end=" ")

while a <= b:

if(a % 2 == 0):

print("{0}".format(a),end=" ")

a = a + 1

print()

7. Write a program to print 1 to 10 using the do-while loop statement.

a = 1

print("Print 1 to 10 using the do-while loop statement:",end=" ")

while True:

print(a,end=" ")

a = a + 1

if(a > 10):

break

print()

8. Write a program to find Armstrong number or not.

a = int(input('Enter a number to check if its armstrong or not: '))

sum = 0

temp = 0

temp = a

while temp > 0:

r = temp % 10

sum += r \*\* 3

temp //= 10

if a == sum:

print(a," is an amstrong number")

else:

print(a," is not an amstrong number")

8. Write a program to find the prime or not.

number = int(input("Enter any number to check prime number or not: "))

# prime number is always greater than 1

if number > 1:

for i in range(2, number):

if (number % i) == 0:

print(number, "is not a prime number")

break

else:

print(number, "is a prime number")

else:

print(number, "is not a prime number")

9. Write a program to palindrome or not.

n = int(input("Enter number to check palindrome or not:"))

temp = n

rev = 0

while(n > 0):

dig = n % 10

rev = rev \* 10 + dig

n = n // 10

if(temp == rev):

print("The number is a palindrome!")

else:

print("The number isn't a palindrome!")

10. Program to check whether a number is EVEN or ODD.

a = int(input('Enter Number to check is EVEN or ODD: '))

if a % 2 == 0:

print("{0} is Even ".format(a))

else:

print("{0} is Odd ".format(a))

PYTHON ARRAY

1. Write a function to add integer values of an array.

arr = [10,20,30,40]

sum = 0

for i in range(0,len(arr)):

sum = sum + arr[i]

print("Sum of all integer values in array: ",sum)

1. Write a function to calculate the average value of an array of integers.

def cal\_average(num):

sum\_num = 0

for i in num:

sum\_num = sum\_num + i

avg = sum\_num / len(num)

return avg

print("The average is", cal\_average([10,21,32,43,54]))

3. Write a program to find the index of an array element.

arr = [1,3,5,3,1,2,6,5,3,8,6,9]

index = arr.index(3)

print("Index of 3: ",index)

index = arr.index(9)

print("Index of 9: ",index)

index = arr.index(8)

print("Index of 8: ",index)

4. Write a function to test if array contains a specific value.

arr = [4,5,45,40,50]

for num in arr:

if num == 5:

print("Element exist")

1. Write a function to remove a specific element from an array.

arr = [44,55,0,15,19,5,4]

arr.remove(0)

print(arr)

1. Write a function to copy an array to another array

arr = ['k','a','s','h','i']

arr1 = []

arr1 = arr.copy()

print(arr1)

7.Write a function to insert an element at a specific position in the array.

arr = [101,303,404,505,606,707,808,909]

arr.insert(1,202) #insert 202 at index 1 in arr

print(arr)

8. Write a function to find the minimum and maximum value of an array.

#Initialize array

arr = [10,3,90,89,56,1,4,5]

minposition = arr.index(min(arr))

print ("The minimum is at position:", minposition)

maxposition = arr.index(max(arr))

print ("The maximum is at position::", maxposition)

9. Write a function to reverse an array of integer values.

arr = [9,8,7,6,5]

arr.reverse()

print(arr)

10. Write a function to find the duplicate values of an array.

arr = [21,11,31,13,11]

for i in range(0,len(arr)):

for k in range(i + 1,len(arr)):

if arr[i] == arr[k]:

print("Duplicate element in given array:",arr[k])

11.Write a program to find the common values between two arrays.

arr = ['k','a','s','h','i']

arr1 = ['s','h','g']

print("Common values in given arrays:",set(arr).intersection(arr1))

12. Write a method to remove duplicate elements from an array.

arr = [11,22,33,11,44,55]

finalarr = [] #empty array

for i in arr:

if i not in finalarr:

finalarr.append(i)

print(finalarr)

13. Write a method to find the second largest number in an array.

arr = [101,404,202,909,606,505,808,303,707]

arr.sort()

print("Second largest number:",arr[-2])

14.Write a method to find number of even number and odd numbers in an array.

arr = [1,2,3,4,5,6,7,8,9]

evennumbers = 0

oddnumbers = 1

for i in arr:

if i % 2 == 0:

evennumbers += 1

else:

oddnumbers += 1

print("Even Numbers in given array:",evennumbers)

print("Odd Numbers in given array:",oddnumbers)

15. Write a function to get the difference of largest and smallest value.

arr = [10,6,11,13,14]

arr.sort() #Sorting in ascending order

print("Diffrence of largest and smallest value:",arr[4]-arr[0])

16. Write a method to verify if the array contains two specified elements(12,23).

arr = [1,12,19,23,33,54]

for i in arr:

if i == 12:

print("Exist in array")

if i == 23:

print("Exist in array")

PYTHON STATICS

1. Define a static variable and access that through a class.

class Python:

staticVariable = 8

print(Python.staticVariable)

Python.staticVariable = 12

print(Python.staticVariable)

1. Change within class

instance = Python()

print(instance.staticVariable) # Gives 12

1. Change within an instance

instance.staticVariable = 15

print(instance.staticVariable)

print(Python.staticVariable)

PYTHON STRING

1. Different ways creating a string.

string = 'Hello'

print(string)

string = "Hello"

print(string)

string1 = '''World'''

print(string1)

string2 = """Welcome to

the world of Python"""

print(string2)

print()

1. Concatenating two strings using + operator

str1 = string + string1

print("Concatenated two different strings:",str1)

print()

1. Finding the length of the string.

print("length of the string:",len(str1))

print()

1. Extract a string using Substring.

str3 = 'manojis'

str1 = 'jis'

str2 = 's'

print("Position of ish:",str3.index(str1))

print("Position of h:",str3.index(str2))

print()

5. Matching a String Against a Regular Expression With matches().

Substr = 'Madara'

str6 = 'Madara once said- Wake up to relity nothing goes as planned in this cursed world'

print(re.match(Substr,str6))

print()

str8 = 'Itachi uchiha'

str1 = 'Madara uchiha'

str2 = str8

print(str8 == str1)

print(str8 == str2)

print(str1 == str2)

print(str8 != str1)

print()

6. startsWith(), endsWith().

string = 'Minato Namikaze'

print(string.startswith("Minato"))

print(string.endswith("kaze"))

print()

1. Trimming strings with strip().

str7 = 'Hello World hi'

print(str7.strip("hi"))

print()

1. Replacing characters in strings with replace()

string = 'Hi World'

print(string.replace("Hi","Hello"))

print()

1. Splitting strings with split()

str9 = 'hi-hello-bye'

print(str9.split("-"))

print()

1. Converting integer objects to Strings.

numb = 10

numb1 = str(numb)

print(numb1)

print(type(numb1))

print()

1. Converting to uppercase and lowercase.

string = 'mansi'

string1 = 'KAPIL'

print(string.upper())

print(string1.lower())

PYTHON INHERITANCE

1. class A():

def display(dp):

print("Display Inside class A ")

2.class A show method

def show(self):

print("Show Inside class A")

3.Inherited or Sub class (Note A in bracket)

class B (A):

def print(pt):

print("Print Inside class B")

class B show method

def show(self):

print("Show Inside class B")

4. Inherited or Sub class (Note B in bracket)

class C (B):

def show(self):

print("Show Inside class C ")

s = A()

s.display()

k= B()

k.print()

g = C()

g.show()

PYTHON ACCESS MODIFIER

1. #super class

class Super:

var1 = None

\_var2 = None

\_\_var3 = None

def \_\_init\_\_(self, var1, var2, var3):

self.var1 = var1

self.\_var2 = var2

self.\_\_var3 = var3

def displayPublicMembers(self):

print("Public Data Member: ", self.var1)

def \_displayProtectedMembers(self):

print("Protected Data Member: ", self.\_var2)

def \_\_displayPrivateMembers(self):

print("Private Data Member: ", self.\_\_var3)

def accessPrivateMembers(self):

self.\_\_displayPrivateMembers()

class Sub(Super):

# constructor

def \_\_init\_\_(self, var1, var2, var3):

Super.\_\_init\_\_(self, var1, var2, var3)

# public member function

def accessProtectedMembers(self):

# accessing protected member functions of super class

self.\_displayProtectedMembers()

# creating objects of the derived class

obj = Sub("KG", 5 , "KG !")

obj.displayPublicMembers()

obj.accessProtectedMembers()

obj.accessPrivateMembers()

print("Object is accessing protected member:", obj.\_var2)

PYHTON ABSTRACT CLASS

1. from abc import ABC, abstractmethod

class Polygon(ABC): #base class / super class

@abstractmethod

def noofsides(self):

pass

class Triangle(Polygon): #subclass

def noofsides(self):

print("Triangle: I have 3 sides")

class Pentagon(Polygon):

def noofsides(self):

print("Pentagon: I have 5 sides")

class Hexagon(Polygon):

# overriding abstract method

def noofsides(self):

print("Hexagon: I have 6 sides")

class Quadrilateral(Polygon): #subclass

# overriding abstract method

def noofsides(self):

print("Quadrilateral: I have 4 sides")

R = Triangle()

R.noofsides()

K = Quadrilateral()

K.noofsides()

R = Pentagon()

R.noofsides()

K = Hexagon()

K.noofsides()

PYTHON FILES

1. Write a program to read text file.

file1 = open("HW.txt","r")

data = file1.read()

print(data)

print()

file2 = open("Blank.txt","w")

str1 = 'Python'

file2.write(str1)

print()

file3 = open("HW.txt","r+")

print(file3.readline(11))

print()

PYTHON CONSTRUCTERS

1.

class A:

# default constructor

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

self.name = "Kashish"

2. a method for printing data members

def print\_A(self):

print(self.name)

3.creating object of the classand method using the bject j

obj = A()

obj.print\_A()

class B(A):

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

self.name = "KG"

def print\_B(self):

print(self.name)

obj1 = B()

obj1.print\_B()

class C:

name = None

\_roll = None

\_\_branch = None

def \_\_init\_\_(self,name,roll,branch):

self.name = name

self.\_roll = roll

self.\_\_branch = branch

def dsiplayName(self):

print("Name:",self.name)

def \_displayRoll(self):

print("Roll:",self.\_roll)

def \_\_displayBranch(self):

print("Branch:",self.\_\_branch)

def access\_\_displayBranch(self):

self.\_\_displayBranch()

class D(C):

def \_\_init\_\_(self,name, roll, branch):

super().\_\_init\_\_(name,roll, branch)

def access\_displayRoll(self):

self.\_displayRoll()

obj = D("Mansi", 5 , "CbSE")

obj.dsiplayName()

obj.access\_displayRoll()

obj.access\_\_displayBranch()

DICTIONARY

#Creating dictionary

Dict = dict([(1,'Kashish'), (2,'Mansi '), (3,'KUHUK'), (4,'Deeps'), (5,'Mukesh')])

print("Dictionary with each item as a pair:",Dict)

2.adding element in dictionary

Dict[6] = 'Nitya'

print("\n Dictionary with new item added:",Dict)

3.updating values in dictionary

Dict[3] = 'Navdisha'

print("\n Dictionary with updated values:",Dict)

print("\n Accessing one value in Dictionary:",Dict[1])

4.deleting value from drictionary

del Dict[6]

print("\n Delete a value from a Dictionary:",Dict)

5.creating a nested dictionary

Dict1 = {1: 'Kashish', 2: 'Kritika',

3:{'Age' : 18, 'Branch' : 'CSE', 'Year' : 'Third Year'}}

print("\n Nested loop Dictionary:",Dict1)

print("\n Accessing an element of a Nested Dictionary:",Dict1[2])

a = [1, 2, 3]

try:

print ("Second element = ",a[1])

# Throws error since there are only 3 elements in array

print ("Fourth element = ",a[3])

except:

print ("An error occurred")

print()

b = [3,2,1]

try:

a == b

except:

print("They are not equal")

else:

print("Both Equal")

print()

try:

k = 5/0

print(k)

except ZeroDivisionError:

print("Can't divide by zero")

finally:

print('This is always executed')