JALATechnologies.com

**Python Basics**

# Write a program to print your name.

print ("Kashish")

# 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")

# 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))

# 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)

# 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)

**Operators**

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

# Store input numbers:

num1 = input('Enter first number: ')

num2 = input('Enter second number: ')

# Add two numbers

sum = float(num1) + float(num2)

# Subtract two numbers

min = float(num1) - float(num2)

# Multiply two numbers

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

#Divide two numbers

div = float(num1) / float(num2)

# Display the sum

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

# Display the subtraction

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

# Display the multiplication

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

# Display the division

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

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

# A sample use of increasing the variable value by one.

a = 0

a += 1

a = a+1

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

# Use of increment operator, here start = 1,step = 1(by default) and stop = 5

print("INCREMENTED FOR LOOP")

for i in range(0, 5):

print(i)

#Use of decrement operator, here start = 5, step = -1 and stop = -1

print("\nDECREMENTED FOR LOOP")

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

print(i)

# 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")

# Program for relational operators (<,<==, >, >==)

a = 4

b = 5

print(a < b) #This operator(<) returns True if the left operand is less than the right operand.

print(a <= b) #This operator(<=)returns True if the left operand is less than or equal to the right operand.

print(a > b) #This operator(>) returns True if the left operand is greater than the right operand.

print(a >= b) #This operator(>=)returns True if the left operand is greater than or equal to the right operand.

print(a == b) #This operator(==)returns True if both the operands are equal i.e. if both the left and the right operand are equal to each other.

print(a != b) #This operator(!=)returns True if both the operands are not equal.

# Print the smaller and larger number.

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

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

#To print larger number

if a > b:

print(a, "is greater ")

else:

print(b, " is greater ")

#To print samller number

if a < b:

print(a, "is smaller2 ")

else:

print(b, " is smaller ")

**Loops**

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

from tkinter import N

#Declaring variable

a = "Bright IT Career"

#using loop to print "Bright IT Career" 10 times

for i in range(10):

print(a)

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

i = 1

#using loop to print 1 to 20 numbers

while(i<=20):

print(i)

i+= 1

3. Program to equal operator and not equal operators.

a = 5

b = 10

print(a ==b) #Equal operator

print(a != b) #Not Equal operator

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

Numbers = [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 Numbers:

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")

9.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")

# if the entered number is less than or equal to 1

# then it is not prime number

else:

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

10. 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!")

# 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))

**Arrays**

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

from audioop import avg

from operator import index

from turtle import position

from typing import final

#Initialize array

arr = [10,20,30,40]

sum = 0

#Loop through the array to calculate sum of elements

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

sum = sum + arr[i]

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

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

def cal\_average(num):

sum\_num = 0

#Loop through the array to average value of elements

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]))

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

#Initialize array

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

#printing element at specific index in array

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)

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

#Initialize array

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

#Loop through array to test if array contains a specific value

for num in arr:

if num == 5:

print("Element exist")

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

#Initialize array

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

#removing a specific element from an array

arr.remove(0)

print(arr)

# Write a function to copy an array to another array

#Initialize array

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

arr1 = [] #creating empy array

arr1 = arr.copy() #copying/duplicating arr in arr1

print(arr1)

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

#Initialize array

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

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

print(arr)

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

#Initialize array

arr = [100,3,3000,20,500,9999,10000,10003]

#minimum value of array

minposition = arr.index(min(arr))

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

#maximum value of array

maxposition = arr.index(max(arr))

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

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

#Initialize array

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

arr.reverse() #to reverse an array of integer values

print(arr)

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

#Initialize array

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

#Using loop to check duplicate values in array

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])

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

#Initialize array

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

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

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

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

#Initialize array

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

finalarr = [] #empty array

#Using loop to remove duplicated elements

for i in arr:

if i not in finalarr:

finalarr.append(i)

print(finalarr)

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

#Initialize array

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

arr.sort() #Sorting in ascending order.

print("Second largest number:",arr[-2]) #displaying the second last element.

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

#Initialize array

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

evennumbers = 0

oddnumbers = 0

#using loop to find even and odd numbers in array

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)

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

#Initialize array

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

arr.sort() #Sorting in ascending order

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

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

#Initialize array

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

#using loop to find if array contains specific elements

for i in arr:

if i == 12:

print("Exist in array")

if i == 23:

print("Exist in array")

**statics**

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

class Python:

# Access through class

staticVariable = 9

print(Python.staticVariable)

2 define Change within an class

Python.staticVariable = 12

print(Python.staticVariable) # Gives 12

3.Define a static variable and access that through a instance

print(instance.staticVariable) # Gives 12

4. Define a static variable and change within the instance

instance.staticVariable = 15

print(instance.staticVariable) # Gives 15

print(Python.staticVariable) #Gives 12

1. Write a class with a default constructor, one argument constructor and two argument constructors. Instantiate the class to call all the constructors of that class from a main class

2. Call the constructors(both default and argument constructors) of super class from a child class

3. Apply private, public, protected and default access modifiers to the constructor

class A:

# default constructor

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

self.name = "Kashish"

# a method for printing data members

def print\_A(self):

print(self.name)

# creating object of the class

obj = A()

# calling the instance method using the object obj

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:

# public data member

name = None

# protected data members

\_roll = None

# private data member

\_\_branch = None

# constructor

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

self.name = name

self.\_roll = roll

self.\_\_branch = branch

def dsiplayName(self):

print("Name:",self.name)

# protected member function

def \_displayRoll(self):

# accessing protected data members

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

# private member function

def \_\_displayBranch(self):

# accessing private data members

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

# public member function

def access\_\_displayBranch(self):

# accessing private member function

self.\_\_displayBranch()

class D(C):

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

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

# public member function

def access\_displayRoll(self):

# accessing protected member functions of super class

self.\_displayRoll()

4. Write a program which illustrates the concept of attributes of a constructor

creating objects of the derived class

obj = D("Kashish", 5 , "CSE")

# calling public member functions of the class

obj.dsiplayName()

obj.access\_displayRoll()

obj.access\_\_displayBranch()

**Inheritance**

A, B and C are classes A is a super class. B is a sub class of A.

C is a sub class of B. Create three methods in each class,

2 methods are specific to each class and third method (override method) should be in all three Classes A, B and C Create a class with main method. Create an object for each class A, B and C in main method and call every method of each class using its own object/instance. Call an overridden method with super class reference to B and C class’s objects

Runtime Polymorphism with Data Members/Instance variables, Repeat the above process only for data member

class A():

def display(dp):

print("Display Inside class A ")

# class A show method

def show(self):

print("Show Inside class A")

# 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")

# Inherited or Sub class (Note B in bracket)

class C (B):

# class C show method

def show(self):

print("Show Inside class C ")

# Driver code

s = A()

s.display()

k= B()

k.print()

g = C()

g.show()

**Dictionary**

. Create a Dictionary with at least 5 key value pairs of the Student ID and Name 1.

1.Adding the values in dictionary

.2. Updating the values in dictionary

3. Accessing the value in dictionary 1.

4. Create a nested loop dictionary 1.

5. Access the values of nested loop dictionary 1.6. Print the keys present in a particular dictionary

7. Delete a value from a dictionary

#Creating dictionary

Dict = dict([(1,'Kashish'), (2,'Kritika'), (3,'Aastha'), (4,'Vaishali'), (5,'Muskan')])

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

#adding element in dictionary

Dict[6] = 'Nitya'

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

#updating values in dictionary

Dict[3] = 'Navdisha'

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

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

#deleting value from drictionary

del Dict[6]

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

#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])