//save files as .py before running otherwise it wont run

Print(“any string”)

//Variable directly as

G

//And input as

G = input(“enter value for G”)

//like that the string in the parenthesis could be blank even…

//functions can be called like this

Pow(5,4)

//in Python Header libraries are called as modules…and to import them…

Import math

//to use its functions…we cant use directly as normal functions…we need to use module name and then //function name just as if math is an object of c++ lol

Math.floor(5.2)

//gives output 5, traditional floor function

//also we could give variables a function , for ex.

Variable1 = math.pow

Variable1(5,4)

//this statement gives us an output which is 625 , 5 power 4

//for concatenating few things…do this

x = input("Enter name:")

print ("hey " + x)

// it will print something like, hey duke , if input to x is given is duke or any other no or string whatever is //the input

//at the end put any input statement so that it wait for user input…i.e. in normal words…getch() of c

Input(“Press<enter>”) //example , this message is printed while it is waiting or input

//or

Input() //this will also do as its just an input we want

// escape sequence or character is same for python…

\ //this is the escape character, you know what it means ;P

// for comment in python…

###any comment you wanna do

//for srting array

Family = [“asd” , “qwe” , “bf”]

//to use for print or anything

Family[0]

//as normal arrays are done…this has “asd”

//also python provides another function…we can also use it from behind anytime we want…

//last element is indexed as -1 then the one before it is indexed as -1…likewise…

Print(Family[-1])

//will print “bf” without the quotes…

//To use an input specifically in new python versions after 3.x since only input() is there and no //raw\_input() function is there…input() takes every thing as string…so if we wanna use the input

//for indexing an array then we cant use a string for an obvious reason…

//so for that we can explicitly make it to int…example

N = int(input(“Enter for n: ”))

//now this would make n an int and the following statement would be a valid one therefore…

Print(Family[n])

//if n =0 then asd is the output

//also in python this is a valid thing…

print('bucky'[3])

print("bucky"[3])

//which gives **output “k”** without double quotes in both of ‘em…

//slicing of a list

temp = [0,2,0,3,0,2,2]

print(temp[2:6])

//will give us an **output as [0, 3, 0, 2]**

print(temp[2:]) //print from 2nd index to end

print(temp[:6]) //print from start to 6th index

print(temp[:]) //complete list

print(temp[-5:-1])

print(temp[-5:]) //from backward -5 to end

print(temp[1:6:2]) //two jumps, i.e. alrenate value, third parameter is jump

print(temp[6:0:-2]) //two jumps and from backwards from backwards range

print(temp[::2]) //two jumps from backwards

//will give respective outputs

[0, 3, 0, 2, 2]

[0, 2, 0, 3, 0, 2]

[0, 2, 0, 3, 0, 2, 2]

[0, 3, 0, 2]

[0, 3, 0, 2, 2]

[2, 3, 2]

[2, 0, 0]

[0, 0, 0, 2]

//check a character

Name = “shreyan”

‘z’ in name

//output False

‘r’ in name

//output True

‘asdf’ in family //here family is the one made above

//output false

‘asd’ in family

//output True

//built in functions like len() …gives no. of elements in a list, i.e. array

Numbers = [1,5,8,0,66,0,4]

Len(numbers)

Max(numbers)

Min(numbers)

List(“shreyan”)

[‘s’,’h’,’r’,’e’,’y’,’a’,’n’]

Del numbers[4] // deletes ‘y’ from that

//methods…

//For a temp used above we can use methods like append() , to append the number given in parenthesis //i.e. add in list…likewise .extend and count…

//another index for numbers above

Numbers.index(5)

//will give u 1 since first time we got 5 in that list is on the 1st index…

//insert , insert any new thing at a new place

Numbers.insert(3,4)

//here 3 represents index and 4 is the value which we have to put there…and other data of the list shifts back to right

//pop with index pops data on that index…remove with the element removes that element …

//reverse reverses the list

// to sort an list having values

Numbers.sort()

//done…

//now for string, as in in a string…without an object is done..like

Sorted(“easyhoss”)

//output every character sorted in a alphabetical manner

// a **Tuple**  in python is like a list but it cannot be modified once made…

//example

Bucky = (15,46,4,92,311)

//the list cannot be altered now therefore

//for srtings..something like this thing can be done…here %s means string type…

bucky= "hey there %s"

>>> abss="qwerty"

>>> print(bucky % abss)

hey there qwerty // this is the **output**

example.find(“there”)

//gives output 4…i.e. there is found at 4th index..likewise….

Str1.join(str2)

//puts str1’s content after every element of str2….

.lower() // puts all letter to lowercase

.replace(“word to be replaced” , “new word”) //just as find and replace

//dictionary in Python is like a 2-d array

Ages = {“dad” : “42” , “mom” : “40”}

Ages[“dad”]

//would give 42

//methods clear clears it completely… copy copies that one to the one it is being assigned… //has\_key[“dad”] , gives True…since dad has a key

//If statement…regular one

tune = 1

>>> if tune == 1:

print("correct")

elseif tune==2 :

print("elseif")

else :

print("else statement")

correct

/ / test statements

's' in 'place'

FALSE

's' in 'place'

TRUE

1 is 2

FALSE

1 is 1

TRUE

and //for &&

or // for ||

//while loop

b = 1

while b< 5 :

print(b)

b+=1

//output 1 to 5

// for loop

gl = ['milk' , 'butter' , 'chesse' , 'bread']

for food in gl :

print('I want ' + food)

//output 4 times output with all the food names in the gl list...

/for dictionary....

name = {'fdsf' : 'sdf','sdsfs' : 'sdgfds','sfsf' : 'dsfadgf'}

for firstname in name :

print(firstname)

// this prints the keys...i.e. fdsf , sdsfs , sfsf

// to print the key's value , i.e. sdf etc...we need to do...name[firstname]

// break keyword for break

// for defining functions...

def function\_name (argument\_1 , argument\_2) : // like wise any number of arguments you //may want to have

// any operation to be performed ...

return s

// or any return thing , if any

// we can put a argument like \*something....what that does is takes multiple number of values as a tuple....

ex.

def sample(\*name) :

print('%s' % name)

// calling of the function sample

sample('qwe','qwe','aasdf')

//to make and pass a dictionary …

Def profile(\*\*items)

Print(items)

//passing a dictionary

Profile(bucky=4,karl=5)

{‘buky’:4,’kalr’:5} //output

Def example(a,b,c):

Return a+b\*c

Tuna=(12,45,78)

Example(\*tuna)

//this would be accepted…similarly we could also pass a dictionary in some other function as parameter

Def example2(\*\*temp):

Print(temp)

hello={‘buky’:4,’kalr’:5}

example2(\*\*hello)

//OOP in Python

//how to create class…

Class classname:

Datamember1=’hello’

Datamember2=’bello’

Def method\_name(Self): // here first parameter is always self

Return ‘first class method’

// object of the class to be created…

firstObject=classname()

//created…now we can access I just like we do….using ‘.’ Operator …

Firstobject.datamember1

//gives ‘hello’

//likewise

Firstobject.method\_name()

//would give ‘first class method’

//self is just like this of c++

//inheritance of classes in python…

Class childclassname(Parentclassname) :

//whatever extra is there of child class

//this is obviously at the time of declaration of child class…

//overriding is the same as in c++…i.e. using the same member name but giving them different values

//multiple inheriting could also be done by using a comma between the parent class names in the //parenthesis...

//making a default constructor in python

Def \_\_init\_\_(self):

//do whatever rubbish u wanna do in ctor

//just as we imported predefined module like math…we can create and import user defined ones so as //to use our user defined functions and variables in any other files asap…

// but for that we need to make that mod file in the default folder of python…since python only imports //from the default folder…

//make a file and save as .py format and then u can import it as such..

//import command..

Import modulename

//to use module things use modulename as object…

//ex

Modulename.var1

//also we could give an alias name for any particular var call or method call…ex

Alias1=modulename.var1

Alias1()

//this gives the exact same result as above…

//one module can be imported only once,….even if we import it more than once then also…any changed made in it wont be reflected back in the place where we are currently using it…

Reload(modname)

//used to reload the module…i..e the changed would be reflected back…

dir(modname)

//this gives us complete list of functions that build in mod contains…

Help(modname)

//gives us complete description of the functions in that mod…including the no of arguments…meaning…etc

Mod.\_\_doc\_\_

//gives us complete documentation of the mod…

**//File Management in python**

Filevariable=open(‘C:/directoryname/subdirectory\_likewise/filename.txt’,’w’)

//here w for writing and r for reading…

Filevariable.write(‘first written string to the file’)

//this will write to the file

//for closing the file when not in use…

Filevariable.close()

Filevariable1.open(‘C:/directoryname/subdirectory\_likewise/filename.txt’,’r’)

New\_variable\_that\_contains\_data\_that\_has\_been\_read=filevariable.read(5)

//here parameter says number of bytes or characters to be read…

//If we dint give any parameter then it reads the complete one i.e form the current cursor to end…

//for line by line transaction…

//Readline() gives, i.e reads till end of a line…i.e. \n…

//Readlines() gives all the lines…

//in write we can put \n character to make new lines….

//the thing we get in is a python list…

//writelines()… we can provide directly a list in it as parameter…