**FYBCA**

**SEMESTER – I**

**BCA1111C03 : Introduction to Programming using Python**

**Date : 16/11/2022**

1. Write a program to read multiple values from a user in single line. And type cast each value in int or float.
2. Write a program to print 1... 10 but break the loop when it reaches to 8.
3. Write a program to print 1...10 skipping the value 6.
4. Create a list that is populated with all the even numbers between 1-1000

Newlist = []

for x in range(1,1000):

if x % 2 == 0:

Newlist.append(x)

print(Newlist)

1. Create a list that is populated with all the odd numbers between 1-1000

Newlist = []

for x in range(1,1000):

if x % 2 != 0:

Newlist.append(x)

print(Newlist)

1. Create a list that is populated with all the numbers divisible by 5 between 1-1000

Newlist = []

for x in range(1,1000):

if x % 5 == 0:

Newlist.append(x)

print(Newlist)

1. Create a list that is populated with square values of numbers between 1-1000

Newlist = []

for x in range(1,1000):

res=x\*\*2

Newlist.append(res)

print(Newlist)

1. Write a program to find sum and mean of only numerical values contained in a list.

**TUPLE :**

1. Create a tuple of your choice.
2. Iterate over the list and display the elements of the tuple.
3. Concatenate two tuples.
4. Create a nested tuple.
5. Iterate over a nested tuple.
6. Demonstrate that the tuple is Immutable.
7. Can we add new element to tuple? If possible, how? Demonstarte.

**SET :**

1. Create a set of your choice.
2. Print the set.
3. Iterate over the set.
4. Demonstrate the use of various methods on a set.

remove()          Removes Element from the Set

add()                adds element to a set

copy()              Returns Shallow Copy of a Set

clear()              remove all elements from a set

discard()          Removes an Element from The Set

pop()                Removes an Arbitrary Element

union()             Returns Union of Sets

update()           Adds Element to The Set.

1. Perform following Set Operations/Functions and Methods

##my\_set = {1,2,3}

##print(my\_set)

##my\_set = {1.0, "Hello", (1, 2, 3)}

##print(my\_set)

##my\_set = {1,2,3,4,3,2}

##print(my\_set)

##my\_set = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1} # singlton set

##print(my\_set)

##my\_set = {} # empty dictionary, not a set

##print(my\_set)

##print(type(my\_set))

##mySet =set() # will create an empty set

##print(type(mySet))

### we can make set from a list

##myList = [1,2,3,4,5]

##mySet = set(myList)

##print(mySet)

##my\_set = set([1,2,3,2])

##print(my\_set)

# we can make set from a tuple.

##myTpl = (1,2,3,4,5,5,5,5,5)

##mySet = set(myTpl)

##print(mySet)

##thisset = {"apple", "banana", "cherry"}

##print(thisset)

##thisset = {"apple", "banana", "cherry"}

##for x in thisset:

##  print(x)

##thisset = {"apple", "banana", "cherry"}

##print("banana" in thisset)

##thisset = {"apple", "banana", "cherry"}

##thisset.add("orange")

##print(thisset)

##thisset = {"apple", "banana", "cherry"}

##print(len(thisset))

##thisset = {"apple", "banana", "cherry"}

##thisset.remove("banana")

##print(thisset)

##thisset = {"apple", "banana", "cherry"}

##thisset.discard("banana")

##print(thisset)

##thisset = {"apple", "banana", "cherry"}

##x = thisset.pop()

##print(x)

##print(thisset)

##thisset = {"apple", "banana", "cherry"}

##thisset.clear()

##print(thisset)

##thisset = {"apple", "banana", "cherry"}

##del thisset

##print(thisset)

##set1 = {"a", "b" , "c"}

##set2 = {1, 2, 3}

##set3 = set1.union(set2)

##print(set3)

##set1 = {"a", "b" , "c"}

##set2 = {1, 2, 3}

##set1.update(set2)

##print(set1)

##thisset = set(("apple", "banana", "cherry")) # note the double round-brackets

##print(thisset)

1. Demonstrate the use of Identity Operators.
2. Demonstrate the use of Bitwise operators.