Name :- Chavada Sanjay

Enrollment = 92010104002 -TD1

0-1

-a Write a program to demonstrate all the basic data type in python

Program.

```
a=20
print("type of",a," is"type(a))

b= 25.3
print ("type of",b," is",type(b))

com = 1+4j
print ("type of",comp," is",type(comp))

name = "Sanjay"
print ("type of",name," is",type(name))

bool = 5>3
print ("type of",bool," is",type(bool))

Set ={"car","bike","truck"}
print (set)

list =["1","22","44"]
print (list)

type=("1","5")
print(type)

C ={1:"Python",2:"Java",3:"C++"}
print(C)
```

lab-1

Output -

type of 20 is <class 'int'>
type of 25.3 is <class 'Float'>
type of (1+4j) is <class 'complex'>
type of Sanjay is <class 'str'>
type of true is <class 'bool'>
{'car', 'bike', 'truck'}
["11", "22", '44']
('1', '5')
{1: 'Python', 2: 'Java', 3: 'C++'}

Algorithm

- Start
- Declare the variable like, a, b, comp, name, etc.
- Now assign the values in that variable.
- Print the result.
- Stop.

Lab-1

-b= Write a program that takes two numbers as command line arguments and prints its summation.

Program:-

```
a= int (input (" Enter First Number : "))
b= int (input (" Enter Second Number : "))
Sum = a+b
print ("The sum of two number is :", sum)
```

Output:

```
Enter First Number: 30
Enter Second Number: 60
The sum of two numbers is: 90
```

Algorithm:

1. Start
2. Take the input From the user
3. Name them a & b.
4. Then do the Summation
5. Sum = a+b
6. print the result of Sum
7. Stop.

Lab-1

1-C Write a program to print the largest and Smallest number of three numbers input from user with and without using library function.

Program : With library function.

```
n1 = int(input("Enter Fist Num: "))
n2 = int(input("Enter Second Num: "))
n3 = int(input("Enter Thirdl num:"))

def largest(n1, n2, n3):
    if (n1 > n2) and (n1 > n3):
        largest = n1
    elif (n2 > n1) and (n2 > n3):
        largest = n2
    else:
        largest = n3
    print("The largest number is:", largest)

def smallest(n1, n2, n3)
    if (n1 < n2) and (n1 < n3):
        Smallest = n1
    elif (n2 < n1) and (n2 < n3):
        smallest = n2
    else
        Smallest = n3
    print("The Smallest number is :", Smallest)
```

Output

```
Enter first number : 56
Enter Second number : 45
Enter third number : 63
The largest number is = 63
The Smallest number is = 45
```

Lab-1

Algorithm

- Start
- Take input from the user
- Then define the library function def
- Then take the if, elif and else. condition.
- print the largest number
- Print the smallest number
- Stop.

* Program - Without library function.

```
n1 = float(input("Enter first num:"))
n2 = float(input("Enter Second num:"))
n3 = float(input("Enter Third num:"))
 if (n1>=n2) and (n1>=n3):
    largest =n3
 elif (n2>=n1) and (n2>=n3):
    largest=n2
 else:
    largest=n3

if (n1<=n2) and (n1<=n3):
    Smallest =n1
 elif (n2<=n1) and (n2<=n3):
    Smallest =n2
 else:
    Smallest =n3

Print ("The largest number is,", largest)
print("The Smallest number is", Smallest)
```

Lab-1

Algorithm :-

- Start
- Take input from user
- Then take a if condition and second if condition.
- Then Print the largest number
- Then print the smallest number
- Stop.

Lab-1

1-a Write a program to calculate GCD of two numbers.

- Program :-

```
import math
a = int(input("Enter number 1 : "))
b = int(input("Enter number 2 : "))
print("The GCD of two number is : " math.GCD(a,b))
```

Output :-

```
Enter number 1 : 10
Enter number 2 : 25
The GCD of two number is : 5
```

Algorithm :-

- Start
- import math
- Take two number input from user a & b.
- print the GCD number.
- Stop.

₹800/-
29/7/2022

Lab-2

Q-a Write a program to calculate the square root of a number by Newton's method.

- Program:-

```
def newton_sqroot(a,a1=100):
    n1 = float(a)
    for i in range(a1):
        a= 0.5 *(a+ a1/a)
    return a
    a = int(input("Enter the number:"))
    print(newton_sqroot(a))
```

Output:

```
Enter the number: 81
9.0
```

- Algorithm
- Start
- Take the function as newton_sqroot.
- Then take a variable with their a & a1 = 100.
- n1 = float(a)
- for i in range(a1):)
- a = 0.5 * (a + n1/a) formula
- Take the input number from user
- print the result.

Lab-2

2-b Write a program for checking whether the given number is an even or not.

- Program:-

```
a = int (input ("Enter the number : "))
if (a % 2) == 0:
    print ("The number is even : ", a)
else
    print ("The number is odd : ", a)
```

Output :-

Enter the number : 4.
The number is even : 4.

Algorithm:-

- Start
- Take the input from user, a.
- if (a % 2 == 0)
- print number is even
- else print number is odd.
- Stop.

Lab-2

Q-C Write a program using while loop that asks the user for a number, and print a countdown that number to 0.

- Program:

```
n = int(input("Enter the number: "));
while n >= 0:
print(n);>
n = n-1;
```

output

Enter the number: 5
5
4
3
2
1
0

Algorithm:

- Start
- Take the input from the user
- Take condition while n >= 0
- Print the number n = n-1 form.
- Stop

Lab-2

Q-d) Write a program that uses for loop to print all the odd numbers in the range input by user.

Program.

```
a= int(input("Enter the lower range:"))
b= int(input("Enter the upper range:"))
for i in range(a, b+1):
    if i in range(a.
        if(i%.2 !=0):
            print(i)
```

Output

```
Enter the lower range : 1
Enter the upper range : 10
1
3
5
7
9
```

- Algorithm:-
- Start
- Take input from the user for upper and lower.
- Then take the loop i in range(a, a+1):
- Take the condition if(i%.2!=0):
- print the odd numbers in the range input.
- Stop.

12/8/2022

Lab-3

3-a Write a program to check wheather given string is palindrome or not.

- Program.

```
num = int(input("Enter a number: "))
a = num
b = 0
while (num>0)
c = num %. 10
  b = b * 10 + c
num = num /10
 if (a==b):
print ("The num is palindrome")
 else:
print ("The num is not palindrome")
```

* Output

Enter a number : 424

The number is palindrome.

- Algorithms.
- Start
- Take the input from user
- Then take the conditions a=num, b=0
- In while loop (num>0)
- c = num %. 10
- b = b * 10 + c
- In if condition (a==b)
- Stop.

Lab-3

3-b Write a program that accepts a String from user and performs the following operations:
(1) Print the String in reverse order.

* Program :-

```
str = input ("Enter a String: ")
str = "Sanjay" [::-1]
print (str)
```

Order Output

Enter a String : Sanjay
yajnas

Algorithm :

- Start
- take - a input from user
- Take -1 for reverse order
- print the output in reverse order.
- Stop

b-3

b-ii Print all the odd indexed characters of String.

Program.

```
Str = input("Enter a String:")
odd_char = []
for a in range(len(Str)):
    if (a%2!=0):
        odd_char.append(Str[a])
print("odd Position characters :", format(odd_char))
```

Output

Enter a String: Sanjay

Sna

Algorithm:

- Start
- Take input from user
- Take the loop in range(len(Str))
- Then take condition if(a%2!=0)
- Then append(Str[a]) for odd characters.
- Print the odd position characters.
- Stop.

Lab->

3-b-iii Print the count of all the vowels in the string.

Program:

```
str = input("Enter a string:")
count = 0
for char in str:
    if char in 'aeiouAEIOU':
        print(char)
        count = count + 1
print("No. of vowels:", count)
```

Output:

```
Enter a string : Sanjay
a
a
No. of vowels : 2
```

Algorithm

- Start
- Take input from user
- Take for loop in str:
- Then using if condition char 'aeiouAEIOU'
- print the character
- print the number of vowels
- print out put
- Stop.

29/8/2022

lab-4

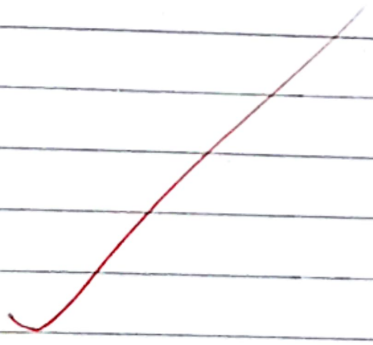
4-A Write a program to create an empty list. Demonstrate the use of the append function to add element onto the list.

Program:-

```
num = []
for i in range(3,15,2)
    num.append(i)
    print(i)
```

Output:-

3.
5
7
9
11
13



Lab-4

4-B Demonstrate the use of the following of list Data. Structure.

i) Operation on list :- copy, count, extend, index, reverse, sort

```
1) #Copy.
   Fruits = ['apple', 'banana', 'cherry', 'Orange']
   copy_fruits = fruits.copy()
   print (copy_fruits)
2) # count
   vowels = ['a', 'e', 'i', 'o', 'i', 'u']
   count = vowels.count(i)
   print ('count':', count)
3) # extend
   language = ['French', 'English', 'German']
   program = ['Java', 'Python']
   language.extend(programing)
   print ('Extended list', language)
4 # index
   print ("Function index')
   nums = [14,5,4,5,7,32]
   x = nums.index(5)
   print(x)
5 # reverse
   print("reverse")
   list = [1,4,3,6,7]
   list.reverse()
   print(list)
6. #.Sort
   vowels = ['e', 'a', 'u', o', 'i']
   print(vowels)
   vowels.sort()
   print ('Sorted list:', vowels)
```

Lab-4

Output :-

['apple', 'banana', 'cherry', 'orange']

count : 2

Extended list : ['French', 'English', 'German', 'Java', 'Python']

I

[7, 6, 3, 4, 1]

['e', 'a', 'u', 'o', 'i']

Sorted list : ['a', 'e', 'i', 'o', 'u']

Lab-4

4-B

ii) Manupulating list: append, insert, pop, remove, clear.

Program :-

1) # append()
print alist = [1,2,3,4]
print(alist)
alist.append(5)
print ("appended list", alist)

2) # insert()
list2 = [1,2,3,4,5,6]
list2.insert(0,7)
print("New List:", list2)

3) #pop()
language = ['Python', 'Java', 'C++', 'Kotlin']
print(language)
print(language.pop(1))
print(language)

4 #Remove()
print list2 = [3,4,1,1,8,9]
print(list2)
list2.remove(4)
print(list2)

5) #clear()
oldlist = ["a", "b", "c", "d"]
newlist = oldlist.clear()
print(newlist)

Lab-4

output

append = appended list : [1,2,3,4,5]

Insert = New List = [7,1,2,3,4,5,6]

Pop = ['Python', 'Java', 'c++', 'Kotlin']
Java
['Python', 'c++', 'Kotlin']

Remove = Before Remove : [3,4,1,1,8,9]
After Remove :- [3,1,1,8,9]

clear = Non.

29/8/2022

Lab-5

A Write a program program to create an empty set. Input the elements from user and write a forloop to add these elements onto the set.

```
Set_a = set()
print("Initial blank set:", set_a)

for x in range(1,5):
    set_a.add(x)
    print(x)
```

Output:

```
Initial blank set: set()
1
2
3
4
```

Algorithm

- Start
- Name a variable
- print the empty set
- add a value using for loop
- print the updated set.
- Stop.

Lab-5

B (i) Demonstrate the use of the following functions of the Set Data Structure:

(1) operations on set: difference(), difference_update(), intersection(), intersection_update(), symmetric_difference(), symmetric_update_difference(), disjoint(), issuperset(), issubset().

```
a = {1, 2, 5, 6, 7}
b = {3, 7, 2, 0, 9}
print("a:", a)
print("b:", b)
print(a.symmetric_difference(b))
print(b.symmetric_difference(a))
print("difference:", a.difference_update(b))
print("difference:", a.difference(b))
print("Intersection:", a.intersection_update(b))
print("Intersection_update:", a.intersection_update(b))
print("symmetric_difference:", a.symmetric_difference(b))
print("Symmetric_difference_upde:", a.symmetric_difference_update(b))
print("isdisjoint:", a.isdisjoint(b))
print("issuperset:", a.issuperset(b))
print("issubset:", a.issubset(b))
```

Algorithm :-

- Start
- make two set and print
- Use, difference(), difference_update(), intersection(), intersection_update(), symmetric_differen(), symmetric_difference_update(), isdisjoint(), issuperset(), issubset() functions for set.
- print all the result.
- Stop.

Lab-5

Output :-

a : {1,2,5,7,8}
b : {0,1,3,7,9}

{0,2,3,5,8,9}
{0,2,3,5,8,9}

difference : {8,2,5}
difference_update : none
intersection : set()
intersection_update : none
Symmetric_difference : {0,1,3,7,9}
Symmetric_difference_update : None
is disjoint : False
is superset : True
issubset : True.

Algorithm
- start
- set the variable
Use the discard()
print the result
add() one element to variable
copy the element
pop the element
Remove the element
Clear the Element in variable
Print all the result

Lab-5

B (ii) Manipulating Set:

discard(), add(), clear(), copy(), pop(), remove()

```
Ourset = {1,5,9}
x = {8,7}
print("our set:", ourset)
print("x:",x)
ourset.add(4)
print("add:", ourset)
new_number = ourset.copy()
print("Copy: ", new_number)
ourset.pop()
print("pop: ", ourset)
ourset.update([3,7])
print("Update:",ourset)
Ourset.discard(9)
print("Discard:",ourset)
ourset.clear()
print("Clear:",ourset)
```

Output:

Ourset: {1,5,9}
x = {8,7}
add: {1,4,5,9}
Copy: {1,4,5,9}
pop: {4,5,9}
Update: {3,4,5,7,9}
Discard: {3,4,5,7}
Clear: set()

Algorithm:

- Start
- Set the variable
- Use the discard(), add(), copy(), pop(), update(), and clear functions.
- Print all the result.

30/9/2022

Lab-6 (A) Write a program to demostrate the use of the following method in touple: (i) count (ii) index

Code

```
thistouple = (1,3,7,6,7,5,4,6,8,5)
x = thistouple.count(5)
print("count :- ",x)
x = thistouple.Index(6)
print("Index :- ",x)
```

Output :-

Count :- 2
Index :- 7

Algorithm :-

Start
create touple with value
use function count()
print count
use function index()
print Index.

Lab-6

b. Create an Empty dictionary and write a program to add Single and multiple elements onto the dictionary.

```
a = dict()
print(a)

b = {'dict1': [1]}
print ("Single Element :-", b)

c = {'dict2': [2,3,4,5,6]}
print ("Multiple Element :- ", c)
```

Output:-

```
{}

Single Element :- {'dict1': [1]}

Multiple Element :- {'dict2': [2,3,4,5,6]}
```

Algorithm

- Create empty dictionary
- print dictionary
- create dictionary with single element
- print the dictionary
- create dictionary with multiple element
- print the dictionary

Lab-6

C-i Write a program demonstrating the use of following of the dictionary data:

(i) Operations on Dictionary: copy(), fromkeys(), get(), item(), keys(), values()

Code():-

```
alphabet = {'a','b','c'}
print("Dictionary:-",alphabet)
New_alphabet = alphabet.copy()
print("Copied:-",New_alphabet)

alphabet = {'a','b','c'}
number = 1
dictionary = dict.fromkeys(alphabets, number)
print("fromkeys()")
print(dictionary)

Car = {"brand": "Ford",
       "Model": "Mustand",
       "Year": 1964}
x = car.get("Price", 15000)
print("get()")
print(x)

x = car.items()
print("Items()")
print(x)

Car x = car.keys()
print("Keys()",x)
```

```
x = car.values()
print("values()")
print(x)
```

Output :-

```
Dictionary :- {'b','a','c'}
Copied :- {'b','a','c'}
fromkeys()
{'b':1, 'a':1, 'c':1}

get()
15000
items()
dict_item([('brand','Ford'),('Model','Mustang'),('year',1964)])
keys()
dict_keys(['brand','Model','year'])
value()
dict_values(['Ford','Mustang',1964])
```

- Algorithm
  - create dictionary with element.
  - print dictionary
  - call method for created dictionary like, copy(), fromkeys(), get(), item(), keys(), value()
  - print all above operations values.

Lab-6

c-ii Manipulating Dictionary : update(), pop(), popitem(), clear()

```
Car = { "Brand": "Ford",
        "Model": "Mustang",
        "year": 1964 }
print("Before popitem", car)
car.popitem()
print("After popitem", car)
print("Before pop", car)
car.pop("Model")
print("After pop", car)
print("Before Update:", car)
car.update({"color": "white"})
print("After update:", car)
numbers = {1: "one", 2: "two"}
print("Before clear", numbers)
numbers.clear()
print("After clear", numbers)
```

Output

- Before popitem {'brand': 'Ford', 'Model': 'Mustang', 'year': 1964}
- After popitem {'Brand': 'Ford', 'Model': 'Mustang'}
Before pop {'Brand': 'Ford', 'Model': 'Mustang'}
After pop {'Brand': 'Ford'}
Before update {'Brand': 'Ford'}
After update {'Brand': 'Ford', 'color': 'white'}
Before clear {1: 'one', 2: 'two'}
After clear {}

Algorithm:

- Create dictionary with multiple keys
- print dictionary
- Use popitem method - car.popitem()
- print dictionary
- Use pop method - car.pop("Model")
- print dictionary
- Use pop update method - and put value that we want to update.
- car.update("color": "white")
- print dictionary
- use clear method for clear a dictionary [car.clear()]
- print dictionary
- it will empty.

9/9/2022

Lab-7

7-A Create a python function to find all unique element in the list.

```
import numpy as np
    def unique(list1):

        x=np.array(list1)
        print(np.unique(x))
list1 = [10,20,10,30,40,40]
 unique(list1)
```

Output :-

[10,20 30 40]

Algorithm:-

- Start
- use numpy library
- create unique function for list using def.
- create array for list
- print the np.unique()
- print the output
- stop.

Lab-7

7-B Create a python function to find all the duplicate elements in a touple

- mylist = (5,3,5,2,1,6,6,4)

```
def dub(myls):
    for i in myls:
        if myls.count(i)>1:
            print(i)
print(dub(mylist))
```

Output:

5
5
6
6
none

Algorithm

- Start
- creat touple with element
- creat dub(myls) using def
- use for loop for ~~mylis~~ i in myls
- use condition if for duplicate element
- print the result
- stop.

Lab-7

7-C Create a program to create a function in python that compares two dictionaries and returns true or false accordingly.

```
dic1 = { 'Name':'Sanjay', 'Age':26 }
dic2 = { 'Name':'Hardik', 'Age':24 }

def dub (dic1, dic2):
    if dic1 == dic2:
        return true
    else:
        return false
print (dub (dic1, dic2)
```

Output:

False

Algorithm:

- Start
- create two dictionaries
- create function dub(dec1, dec2) using def
- Use condition for dic1 and dic2
- if dic1 == dic2
  call return true
- else and if it not dic1 == dic2
  call return false
- print output.
- Stop.

Rec 23/9/2022

Lab-8

A:- Write a program to demostrate recursion in python.

```
def printpattern(targetNumber):
    if (targetNumber <=0):
        print (targetNumber)
        return
    print (targetNumber)
    printpattern (targetNumber-5)
    print (targetNumber
n=10
printpattern(n)
```

Output:-

```
10
5
0
5
10
```

Algorithm

- Start
- define function - printpattern (targetNumber)
- Create Condition that if targetNumber <=0
- print targetNumber
- return
- Print targetNumber
- printpattern and minus five from targetNumber
- Print targetNumber
- assign value in variabel n = 10
- printpattern (n)
- STOP.

Lab-8

8-b Create a function for stack data structure in python and implement necessary operations.

```
stack = ['a','b']
print('Initial stack']
print (stack)
stack.append('c')
print("After Append", stack)
print("\nElements poppeed from stack:')
print (stack.pop())
print('\n stack after elements are popped:')
print(stack)
```

Output :-

Initial stack ['a','b']
After Append ['a','b','c']
Element popeed from stack : c
stack Element after popped : ['a','b']

Algorithm :-

- Start
- Create list with elemement a & b for stack.
- print stack
- append c in stack (stack.appened('c')
- print stack
- pop an Element from stack. (stack.pop())
- print stack.
- Stop.

Lab-8

8-C Create a function for queue data structure in python and Implement necessary operation.

```
class Queue:
    def __init__(self):
        self.queue = []
    def enqueue(self, item):
        self.queue.append(item)
    def dequeue(self):
        if len(self.queue) < 1:
            return none
        return self.queue.pop(0)
    def display(self):
        print(self.queue)
    def size(self):
        return len(self.queue)
q = Queue()
q.enqueue(1)
q.enqueue(2)
q.enqueue(3)
q.enqueue(4)
q.enqueue(5)
q.display()
q.dequeue()
print("After removing an element")
q.display()
```

Lab-8 C

Algorithm :-

Output :-

[1, 2, 3, 4, 5]

After removing an element

[2, 3, 4, 5]

Algorithm :-

- Create class name queue
- create function __init__ with self object. using def
- creat list
- create function enqueue (self, item) with def
- append item (self.queue.append(item))
- create function dequeue (self) using def.
- make condition if length of queue < 1:
- then return None
- Or Return self.queue.pop(0) [
- pop element in queue index 0]
- create function display (self) using def.
- print (self.queue)
- create function size (self): using def
- return length of list (queue)
- make q = Queue.
- append 1 to 5 item in queue using enqueue.
- display queue.
- pop item from queue using dequeue
- print after removed element.
- display the queue.
- Stop.