**ame=["man","bull","cow"]**

**print(li\_name[-2])**

**li\_name[-2]="cat"**

**print(li\_name[-2])**

**print(li\_name)**

**li\_name.extend(["tiger"])**

**print(li\_name)**

**li\_name.insert(3,"lion")**

**print(li\_name)**

**print(li\_name[1:3])**

**print(li\_name[:3])**

**print(li\_name[1:])**

**print( li\_name.index("cat"))**

**try:**

**l1 = li\_name.index("lion")**

**except:**

**l1 = "mouse not found"**

**print(l1)**

**for i in li\_name:**

**print(i.upper())**

**ind=0**

**while ind < len(li\_name) :**

**print(li\_name[ind])**

**ind +=1**

**#sorting an string list**

**print(li\_name)**

**li\_name.sort()**

**print(li\_name)**

**#concatinate two strings**

**l1\_name=["mango","apple","grapes","banna"]**

**l2=li\_name+l1\_name**

**print(l2)**

**# finding lenght of string**

**print(len(l1\_name))**

**print(len(li\_name))**

**print(len(l2))**

**# add new contet on a list or string**

**li\_name.append("hello")**

**print(li\_name)**

**# to print a perticuler range using for loop**

**for i in range(5):**

**print(i)**

**print("next part")**

**# to skipp the number in range**

**for i in range(1,20,3):**

**print(i)**

**print("next part")**

**# also in decresing order**

**for i in range(20,10,-3):**

**print(i)**

**#now in list**

**for i in range(0,len(l2),2):**

**print(l2[i])**

**print("next part")**

**#simple dictonary creation**

**dict\_name={"2618":"parth",**

**"2637":"yogesh",**

**"2719":"viraj",**

**"2603":"aditya"}**

**print(dict\_name)**

**d1= dict\_name["2618"]**

**print(d1)**

**d2=dict\_name["2637"]**

**print(d2)**

**#set value using key**

**#printing dictonary before changing value**

**print(dict\_name)**

**dict\_name["2618"]="nayan"**

**d3=dict\_name["2618"]**

**print(d3)**

**#printing dictonary after changing value**

**print(dict\_name)**

**#add new key and its associated value in dictonary ----- len() also works in dictonaries**

**dict\_name["2696"]="pradyam"**

**d4=dict\_name["2696"]**

**print(d4)**

**print(len(dict\_name))**

**# remove key and its associted value in dictonary using (del)**

**del dict\_name["2637"]**

**print(dict\_name)**

**print(len(dict\_name))**

**#in dictonary data type of or data structure of value is different**

**dict\_1={"list\_1":["male","female","others"],**

**"str\_1":"hello"}**

**print(dict\_1)**

**#using for loop accesing dictionary one keys multiple elements**

**for i in dict\_1["list\_1"]:**

**print(f" {i}")**

**#to find if abc key is present in dictonry using keys() also print a elemnts by specifyning its index number**

**if "list\_1" in dict\_1.keys():**

**print(f"elements is :{dict\_1['list\_1'] [1]}")**

**# if not**

**if "2611" in dict\_name.keys():**

**print(f"2611 is {dict\_name['2611']}")**

**else:**

**print("no student exist in this roll number")**

**# if there is list in dictonary then to finding the presense of value in the list key using values()**

**print("viraj" in dict\_name.values() )**

**#nested dictonaries**

**dict\_info={**

**"parth":{**

**"phon\_number": "8530611635",**

**"email\_id":"parthmahadik32@gmail.com"**

**},**

**"chinmay":{**

**"phone\_number":"9209885875",**

**"email\_id":"chinmaykhatu@22.com"**

**}**

**}**

**print(dict\_info)**

**d5=dict\_info["parth"]**

**print(d5)**

**#similer for chinmay**

**#tuple creation**

**tup\_d=("mon","tues","wed","thur","fri","sat")**

**print(tup\_d[2])**

**#using for loop we can print all the day or oneof the day using if staement**

**for day in tup\_d:**

**print(day)**

**if "we" in tup\_d:**

**print("yes")**

**else:**

**print("no")**

**#we cannot chnge value in tuples if try it gives errors to us**

**try:**

**tup\_d[2]="new wed"**

**except:**

**print("no didn't change ")**

**#similarly u cant remove any elements in tuple u need to del entire tuple**

**try:**

**del tup\_d[2]**

**except:**

**print("no can't able to remove")**

**#del tup\_d**

**#print(tup\_d)# here we got an error**

**#conversion of list into tuple**

**li\_1=[1,2,3,4,5]**

**li\_2=tuple(li\_1)**

**print(li\_1)**

**print(li\_2)**

**print(type(li\_1))**

**print(type(li\_2))**

**#asssignment of tuple values to different variable**

**info=("8530611635","parthmahai@33gamil.com")**

**(phone,email)=info**

**print(f"phone:{phone}")**

**print(f"email:{email}")**

**#using function**

**def hight(numbers):**

**highest=max(numbers)**

**lowest=min(numbers)**

**return(highest,lowest)**

**list\_numbers=[26,11,44,78,90,76]**

**(highest,lowest)=hight(list\_numbers)**

**print(f"high height is :{highest}")**

**print(f"low height is :{lowest}")**

**#using loops**

**contact=[("parth")]**

**oop concept in python**

**#oop notes**

**# creating a simple class and its object**

**#objects are use first instant attributes rather than class attributes**

**class first:**

**print("hello I am Parth")**

**def greet(self):**

**print("good morning")**

**f=first()**

**f.greet()**

**#use @staticmethod to reduse to give self prarameter**

**class second:**

**@staticmethod**

**def greet():**

**print("hello i am independet of self")**

**# contructor are use to initikize the object automatically whenever object is created**

**class third:**

**def \_\_init\_\_(self, name):**

**self.name=name**

**print(name)**

**t=third("i am python")**

**#example of class & objects**

**class worker:**

**def \_\_init\_\_(self,name,salary):**

**self.name=name**

**self.salary=salary**

**def showinfo(self):**

**print(f"name of employe:{self.name} ")**

**print(f"salary of employee:{self.salary}")**

**p=worker("parth",50000)**

**c=worker("chinmay",80000)**

**p.showinfo()**

**print("\n")**

**c.showinfo()**

**# inheritance is way to creat a new class form existing class**

**class employee:**

**company="google"**

**print("i am a employee")**

**class prog(employee):**

**#company="youtube"**

**print("hello i am prog")**

**e=employee()**

**print(e.company)**

**p=prog()**

**"""**

**prog class is inherit with emplyee**

**when attributes is not preseent in child class ( e.g-prog)**

**then it goes to cheack attributes of employee class**

**"""**

**print(p.company)**

**# type 1 inheritance - single inheritance**

**class e1: # parent class**

**com = "TATA"**

**em\_name="parth"**

**class e2: # child class**

**com = "infosys"**

**em\_name="chinmay"**

**e\_1=e1()**

**print(e\_1.com)**

**print(e\_1.em\_name)**

**e\_2=e2()**

**print(e\_2.com)**

**print(e\_2.em\_name)**

**# type 2 - mutiple ineheritance**

**class p1: # parent class**

**com = "TATA"**

**em\_name="parth"**

**class p2: # child class**

**com = "infosys"**

**em\_name="chinmay"**

**class p3(p1,p2):**

**#com="google"**

**em\_name="yogesh"**

**p\_1=p1()**

**print(p\_1.com)**

**print(p\_1.em\_name)**

**p\_2=p2()**

**print(p\_2.com)**

**print(p\_2.em\_name)**

**p\_3=p3()**

**print(p\_3.com)**

**"""**

**after removing com attributes from child class (i.e class p3 )**

**it use parent class attributes**

**"""**

**print(p\_3.em\_name)**

**# type 3 multilevel inheritance**

**class w1: # parent class**

**com = "TATA"**

**em\_name="parth"**

**class w2(w1): # child class**

**#com = "infosys"**

**em\_name="chinmay"**

**class w3(w2):**

**#com="google"**

**em\_name="yogesh"**

**w\_1=w1()**

**print(w\_1.com)**

**print(w\_1.em\_name)**

**w\_2=w2()**

**print(w\_2.com)**

**print(w\_2.em\_name)**

**"""**

**if we remove parent attributes of child class (i.e class w2) then it inherent**

**grand parent class attributes(i.e class w1)**

**"""**

**w\_3=w3()**

**print(w\_3.com)**

**print(w\_3.em\_name)**