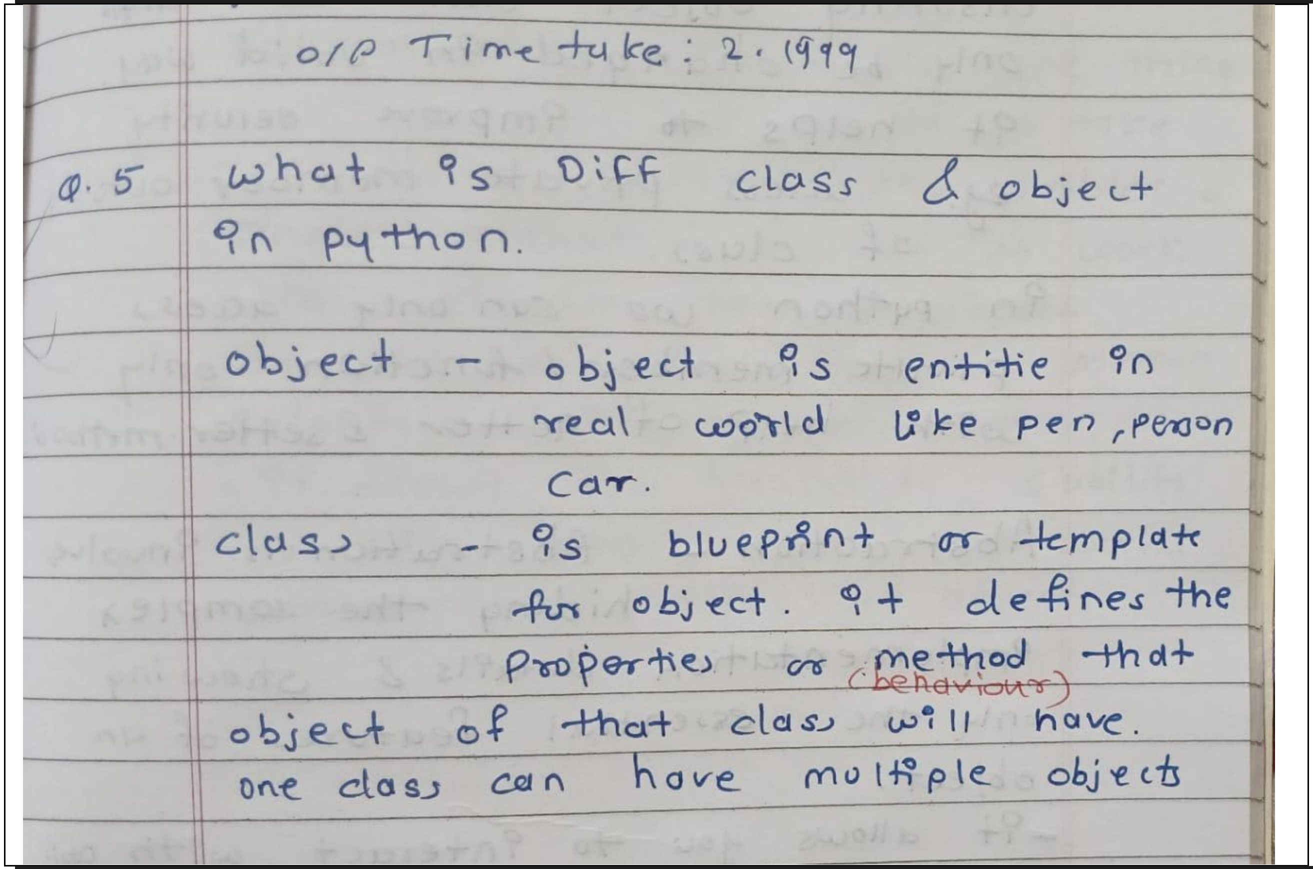
```
# Section A: Theory (10 Questions)

# Q1. What is slicing in python? Brief it.
# slicing: referse extracting part of an sequentiol element


# Q2. What is Keywords in python?
# keywords are reservered word in python they desined for specific use.
# like in True False ,in, else, if we can't use keyword names as varibales  function  names or
class names


# Q3. How can you remove duplicates from a list? Provide at least two methods.
# 1) with help of set method we can remove duplicated in set
# l=[1,2,3,4,1,2,3,4,1,2,3,4,8,9,0,10,203,40]
# l=list(set(l))
# print(l) #--> [0, 1, 2, 3, 4, 8, 9, 10, 203, 40]
# here set is stores only unique values hence list is convered into set.

# 2) with help if terating over list and using conditionals statements
# l=[1,2,3,4,1,2,3,4,1,2,3,4,8,9,0,10,203,40]
# u_l=[]
# for i in l:
#     if i not in u_l:
#         u_l.append(i)
# print(u_l)#--> [0, 1, 2, 3, 4, 8, 9, 10, 203, 40]


# Q4. What are the main characteristics of a tuple in Python?
# the main characteristics of tuple is immutablity
# we can't change tuple elements once created
# t=(1,2,3)
# t[-1]='p'# Type_Error tuple not support item assignment
```

# Q5. What is the difference between a class and an object in Python?



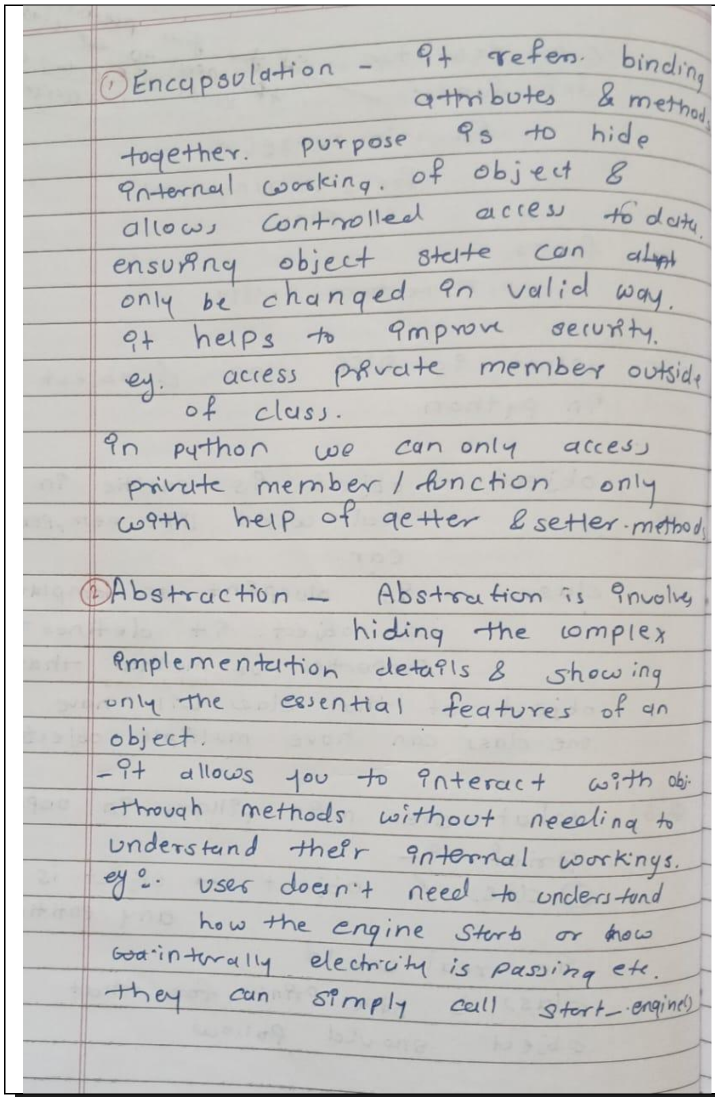
o/p Time take : 2.1999

Q. 5 what is Diff class & object in python.

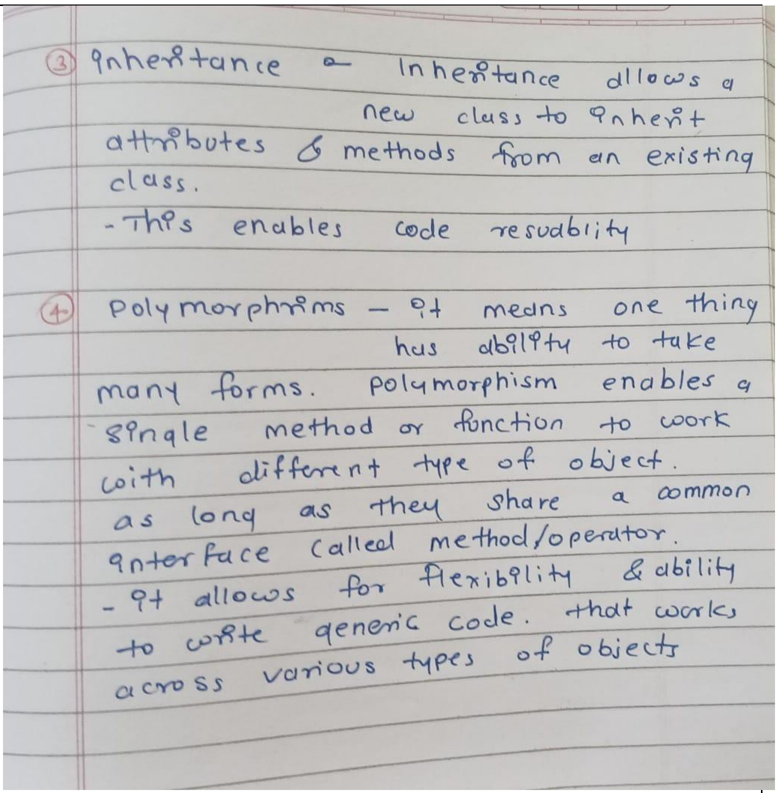
object - object is entitie in real world like pen, person car.

class - is blueprint or template for object. it defines the properties or method (behaviour) that object of that class will have. one class can have multiple objects

# Q6. What are main pillars in OOPS. Brief it.


① Encapsulation - It refers binding attributes & methods together. purpose is to hide internal working of object & allows controlled access to data ensuring object state can ~~alapt~~ only be changed in valid way. It helps to improve security.
eg. access private member outside of class.
In python we can only access private member / function only with help of getter & setter method.

② Abstraction — Abstraction is involves hiding the complex implementation details & showing only the essential features of an object.
- It allows you to interact with obj. through methods without needing to understand their internal workings.
eg:- user doesn't need to understand how the engine starts or how internally electricity is passing etc. they can simply call start_engine()


③ Inheritance - Inheritance allows a new class to inherit attributes & methods from an existing class.
- This enables code resuablity

④ Polymorphisms — It means one thing has ability to take many forms. polymorphism enables a single method or function to work with different type of object. as long as they share a common interface called method/operator.
- It allows for flexibility & ability to write generic code. that works across various types of objects

```
# Q7. What is negative indexes and why are they used?
# python negative index starts from -1 it extract elements from last
# when we have to travers reversly in that case we use negative indexing


# Q8. What are Python’s mutable and immutable types?
# immutable: it means once  data is created we can't change/modify original data
# types like: tuple,string,number,frozenset
#mutable: it means once a data is created we change or modify original data
# types like: list,set,dictionary
# Q9. Explain Python’s list comprehensions.
# Code Readability: if provide more readable way to create a list insted of using loop and
conditions in multiple lines
# effieneccy: compaire to loop it is faster and use less memory
# e.g
# import timeit
# l=list(range(100000))

# time_comprehension = timeit.timeit('[i for i in l]', globals=globals(),number=1000)
# time_comprehension2 = timeit.timeit('''
# l2 = []
# for i in l:
#     l2.append(i)
# ''', globals=globals(), number=1000)
# print(time_comprehension)
# print(time_comprehension2)





# Q10. What is difference between module and a package?
```

Q 10 what is Diff bet^n module & package

*module - is single file eg. (file.py)

package - is directy containing multiple modules. or othe subdirectories like shape

Circle.py

rectandle.py

module :- a module that contains functions class variable. allows related functionality together.

package :- must contain _init__.py file which can be empty or contain initilization code.

module :- structure of file in module is One $\boxed{\text{.py file}}$

Package :- structure of folder in ~~[illegible]~~ package is multiple .py file & one _init__.py file.

module :- import module_name

eg import math

from math import sum

Package :- import package_name. module name

or

from package-name import module-name

from numpy import random

# Section B: Correct the Code (10 Questions)

```
# Q1.
# num = -5
# if num < 0:
#     print("Negative")
# elif num == 0
#     print("Zero")
# else
#     print("Positive")
# ANS
# if num < 0:
#     print("Negative")
# elif num == 0:
#     print("Zero")
# else:
#     print("Positive")


# Q2.
# year = 2024
# if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:
# print(f"{year} is a leap year.")
# else:
# print(f"{year} is not a leap year.")
# ANS
# if (year % 4 == 0):
#    if year%100==0:
#       if year%400==0:
#             print(f"{year} is a leap year.")
#       else:
#           print(f"{year} is not a leap year.")
#    else:
#        print(f"{year} is a leap year.")
# else:
#    print(f"{year} is not a leap year.")
# Q3.
# a = 10
# b = 20
# c = 15
# if a > b and c:
#     print("The highest number is", a)
# elif b > a and c:
#     print("The highest number is", b)
# else:
#     print("The highest number is", c)
# ANS
# a = 10
# b = 20
# c = 15
# if a > b and a>c:
#     print("The highest number is", a)
# elif b > a and b>c:
#     print("The highest number is", b)
# else:
#     print("The highest number is", c)



# Q4. What will be the output of this code?
# x = 10
# for i in range(x):
#       if i % 2 == 0:
#           continue
#       print(i)
# Output:  1,3,5,7,9


# Q5.
# for i in range(1, 21):
#   if i % 2 == 0:
#     print(i)
# output: all even numbers from  to 20
# Q6.
# num = 1234
# total = 0
# while num > 0:
#     total += num % 10
#     num = num // 10
# print("Sum of digits:", total)
# output: 10


# Q7.
# score = int(input("Enter your score: "))
# if score >= 90:
#     print("Grade: A")
# elif score >= 80:
#     print("Grade: B")
# elif score >= 70:
#     print("Grade: C")
# elif score >= 60:
#     print("Grade: D")
# else:
#     print("Grade: F")
# output: Enter your score: 55
# Grade: F



# Q8.
# rows = 5
# for i in range(1, rows + 1):
#     for j in range(1, i + 1):
#         print("*")
# ANS:
# rows = 5
# for i in range(1, rows + 1):
#     for j in range(1, i + 1):
#         print("*")
#     print()
# Q9.
# class Car:
#     def __init__(self, model):
#         self.model = model
#     def describe(self):
#         print("This is a", self.model)

# car = Car("Toyota")
# car.describe()
# Output: This is a Tohoto


# Q10.
# class Animal:
# def __init__(self, name):
#     self.name = name
# def speak(self):
#     print("Animal speaks")

# class Dog(Animal):
# def speak(self):
#     print("Woof")

# dog = Dog("Buddy")
# dog.speak()
# Section C: Write Code For (10 Questions)
# Q1. Write a Python function to count the frequency of each word in a given text document
and return a dictionary with word frequencies.
# s=input("Enter string ")
# word=[]
# p=''
# print(s[len(s)-1])
# for i in range(len(s)):

#     if s[i]!=" " and i!=len(s)-1:
#         p=p+s[i]
#     else:
#         if i==len(s)-1:
#             word.append(p+s[-1])
#         else:
#             word.append(p)
#         p=''
# print(word)
# d={}
# for i in word:
#     if i in d:
#         d[i]+=1
#     else:
#         d[i]=1
# print(d)
# output:
# Enter string am lavanya lnya am
# ['am', 'lavanya', 'lnya', 'am']
# {'am': 2, 'lavanya': 1, 'lnya': 1}


# Q2. Write a lambda function that returns "Even" if a number is even and "Odd" if it is odd.
# print((lambda x: "Even" if x%2==0 else "Odd")(10))
# output: Even


# Q3. Write a Python program to create a dictionary that stores student names as keys and
their scores as values.
#     Write a function that returns the name of the student with the highest score.
# d={name:value for name,value in zip(list(map(str,input("enter names of students ").split(" "))),list(map(int,input("enter marks  of respective students ").split(" "))))}
# def highest_marks(d):
#     maxi=max(d.values())
#     for i,j in d.items():
#         if j==maxi:
#             print("Higesh Score got by:",i,j)
# highest_marks(d)
# output:
# enter names of students lavanya samarth vaishnavi megha meera puri
# enter marks  of respective students 10 20 30 40 50 50
# Higesh Score got by: meera 50
# Higesh Score got by: puri 50
```

# Q4. Develop a function that finds the word with the maximum frequency in a given text document and returns the word along with its frequency.

```
# s=input("Enter string ")
# def Frq(s):
#   word=[]
#   p=''

#   for i in range(len(s)):

#     if s[i]!=" " and i!=len(s)-1:
#         p=p+s[i]
#     else:
#         if i==len(s)-1:
#             word.append(p+s[-1])
#         else:
#             word.append(p)
#         p=''

#   d={}
#   for i in word:
#     if i in d:
#         d[i]+=1
#     else:
#         d[i]=1
#   print(d)
# Frq(s)
# output:
# nter string am lavanya am
# {'am': 2, 'lavanya': 1}


# Q5. Write a Python program using filter() to extract words with more than 5 characters from
a list.
#     # Input: ["apple", "banana", "cat", "elephant"]
#     # Output: ["banana", "elephant"]
# l=["apple", "banana", "cat", "elephant"]
# print(list(filter(lambda x:len(x)>5,l)))
# output:
# ['banana', 'elephant']


# Q6. Write a Python program using reduce() to find the maximum number in a list.
#     # Input: [3, 7, 2, 8, 5]
#     # Output: 8
# from functools import reduce
# l=[3, 7, 2, 8, 5]
# print(reduce(lambda x,y: x if x>y else y,l))
# output: 8


# Q7. Create a dictionary from two lists without using zip function.
# score=[10,20,30]
# name=['A',"B","E"]
# d={}
# for i in range(len(name)):
#     d.update({name[i]:score[i]})
# print(d)
# output:{'A': 10, 'B': 20, 'E': 30}


# Q8. Given two dictionaries, write a function to find and return a new dictionary containing
only the common keys and their corresponding values.
# d1={'A':10,'B':20,'C':30,'D':9}
# d2={'AA':10,'B':200,'CC':30,'D':90}
# mer={}
# for i in d1:
#     if i in d2:
#         mer.update({i:[d1[i],d2[i]]})
# print(mer)
# output:
# {'B': [20, 200], 'D': [9, 90]}


# Q9. Write a Python class called Circle that has a method to calculate the area and the
circumference of the circle.
# class Circle:
#     def __init__(self,r):
#         self.r=r
#     def area(self):
#         print(3.14*self.r*self.r)
#     def circumference(self):
#         print(2*(3.14*self.r))
# c=Circle(4)
# c.area()
# c.circumference()


# Q10. Create a class BankAccount that has the following properties and methods:
#      balance (initially set to 0)
#      deposit(amount) method that adds the amount to the balance
#      withdraw(amount) method that subtracts the amount from the balance
#      display_balance() method to show the current balance
#      A validation check to ensure that withdrawal does not exceed the available balance.
# class BankAccount:
#     def __init__(self):
#         self.bal=0
#     def deposit(self):
#         amt=int(input("Enter amount "))
#         self.bal+=amt
#         print("Deposie Succeful")
#     def withdraw(self):
#         amt=int(input("enter amount "))
#         if self.bal<amt:
```

```
#             print("Insufficient Balance")
#         else:
#             self.bal-=amt
#             print("Withdrwal suffecful")
#     def display_balance(self):
#         print("Current Balance:",self.bal)
# b=BankAccount()
# b.display_balance()
# b.deposit()
# b.display_balance()
# b.withdraw()
# b.display_balance()

# # output:
# # Current Balance: 0
# # Enter amount 100
# # Deposie Succeful
# # Current Balance: 100
# # enter amount 20
# # Withdrwal suffecful
# # Current Balance: 80
```