TASK 1

class EmployeeClass:

def \_\_init\_\_(self, emp\_name,emp\_id ,emp\_salary):

self.name = emp\_name

self.id = emp\_id

self.salary = emp\_salary

emp\_obj = EmployeeClass('Sam', 192966,50000)

print(emp\_obj.name, emp\_id,emp\_obj.salary)

/////////////////////////////////////////////////

TASK 2

class EmployeeClass:

@classmethod

def store\_company\_name(self,cname):

cls.company\_name=cname

@classmethod

def get\_company\_name(cls):

return cls.company\_name

@classmethod

def store\_company\_id(self,cId):

cls.company\_Id=cId

@classmethod

def get\_company\_id(cls):

return cls.company\_id

@classmethod

def store\_company\_salary(self,csalary):

cls.company\_salary=csalary

@classmethod

def get\_company\_salary(cls):

return cls.company\_salary

E1= EmployeeClass()

E2= EmployeeClass()

E3= EmployeeClass()

E1. store\_company\_name(“cmp-1”)

E2. store\_company\_id(100)

E3. store\_company\_salary(1000)

print(E1. get\_company\_name(), end="\n\n")

print(E2. get\_company\_id(), end="\n\n")

print(E3. get\_company\_salary(), end="\n\n")

//////////////////////////////////////////////////////

TASK 3

class EmployeeClass:

@classmethod

def store\_company\_name(self,cname):

cls.company\_name=cname

@classmethod

def get\_company\_name(cls):

return cls.company\_name

@classmethod

def store\_company\_id(self,cId):

cls.company\_Id=cId

@classmethod

def get\_company\_id(cls):

return cls.company\_id

@classmethod

def store\_company\_salary(self,csalary):

cls.company\_salary=csalary

@classmethod

def get\_company\_salary(cls):

return cls.company\_salary

companyName = "XYZ Company"

companyAddress= "XYZ address"

E1= EmployeeClass()

E2= EmployeeClass()

E3= EmployeeClass()

companyName+E1.store\_company\_name(“emp-1”)+companyAddress

companyName+E2.store\_company\_id(100)+companyAddress

companyName+E3.store\_company\_salary(1000)+companyAddress

def my\_company\_decorator(my\_func):

def decorated\_func(companyName+E1.store\_company\_name(“emp-1”)+companyAddress, companyName+E2.store\_company\_id(100)+companyAddress, companyName+E3.store\_company\_salary(1000)+companyAddress):

print("Result Is:")

my\_func(companyName+E1.store\_company\_name(“emp-1”)+companyAddress, companyName+E2.store\_company\_id(100)+companyAddress, companyName+E3.store\_company\_salary(1000)+companyAddress)

print("End Of The Result")

return decorated\_func

print(E1. get\_company\_name(), end="\n\n")

print(E2. get\_company\_id(), end="\n\n")

print(E3. get\_company\_salary(), end="\n\n")

/////////////////////////////////////////////////////////////////////////////////////////

TASK 4

class EmployeeClass:

class EmployeeClass:

def \_\_init\_\_(self, emp\_name,emp\_id ,emp\_salary):

self.name = emp\_name

self.id = emp\_id

self.salary = emp\_salary

my\_string = EmployeeClass(‘emp-1’, 100, 1000)

my\_string= emp\_obj.name+emp\_id+emp\_obj.salary

for i in my\_string:

print("Each Char:", i)

print("\n\n")

/////////////////////////////////////////////////////////////

TASK5

class Companyheadname:

def set\_headname(self, company):

self.company = company

def get\_company(self):

return self.company

class Companygetname:

def set\_headname(self, company):

self.company = company

def get\_company(self):

return self.company

class EmployeeClass(Companyheadname, Companygetname):

def set\_name(self, ename):

self.ename = ename

def get\_name(self):

return self.ename

e1 = EmployeeClass()

print("Company headname:", e1.get\_company())

///////////////////////////////////////

Task 6

class Employee:

def add\_salary(self, salary):

self.salary = salary

def get\_salary(self):

return self.salary

@staticmethod

def compute\_avg\_salary(sal1, sal2):

return (sal1+sal2)/2

e1 = Employee()

e2 = Employee()

e1.add\_salary(20000)

e2.add\_salary(30000)

s1 = e1.get\_salary()

s2 = e2.get\_salary()

avg = e1.compute\_avg\_salary(s1,s2)

print("Avg Sal:", avg)

//////////////////////////////////////////////////////////////

Task 7 and Task 8

Due to time constraint , I could not complete

////////////////////////////////////////////////////////////////