LAB 6: Hướng Đối tượng trong Python

I. Ôn lại lý thuyết

- **Đối tượng (object)**
  Đối tượng là một thực thể có trạng thái và hành vi. Nó có thể là bất kỳ đối tượng trong thế giới thực như chuột, bàn phím, ghế, bàn, bút, v.v.
  Hàm **__init__()** tích hợp sẵn. Tất cả các lớp đều có một hàm gọi là __init__(), hàm này luôn được thực thi khi lớp đang được bắt đầu.
- **Lớp (class)**
  Lớp có thể được định nghĩa là một tập hợp các đối tượng. Nó là một thực thể logic có một số thuộc tính và phương thức cụ thể.
  Ví dụ: có một lớp nhân viên thì nó phải chứa một thuộc tính và phương thức, tức là một địa chỉ, tên, tuổi, lương, v.v.
- **Phương thức (Method)**
  - Phương thức là một hàm được liên kết với một đối tượng.
  - Một phương thức không phải là duy nhất cho các thể hiện của lớp.
- **Kế thừa (Inheritance)**
  Xác định rằng đối tượng con có được tất cả các thuộc tính và hành vi của đối tượng cha. Tạo một lớp sử dụng tất cả các thuộc tính và hành vi của lớp khác.
  Lớp mới được biết đến như là một lớp dẫn xuất hoặc lớp con và lớp còn lại gọi là lớp cơ sở hoặc lớp cha. Kế thừa giúp tái sử dụng lại mã nguồn.
- **Đa hình (Polymorphism)**
  Poly có nghĩa là nhiều và Morphs có nghĩa là hình thức, hình dạng. Bằng đa hình, chúng ta hiểu rằng một nhiệm vụ có thể được thực hiện theo những cách khác nhau.
- **Đóng gói (Encapsulation)**
  Đóng gói cũng là một khía cạnh quan trọng của lập trình hướng đối tượng. Nó được sử dụng để hạn chế quyền truy cập vào các phương thức và biến. Trong đóng gói, mã và dữ liệu được gói cùng nhau trong một đơn vị.
- **Trừu tượng (Abstraction)**
  Trừu tượng hóa dữ liệu và đóng gói cả hai thường được sử dụng như từ đồng nghĩa. Cả hai đều gần như đồng nghĩa vì sự trừu tượng hóa dữ liệu đạt được thông qua việc đóng gói.

Ví dụ:

- Tạo 1 Class, với thuộc tính là q =10,

```
class MyClass:
q = 10
```

- Tạo 1 Object:

```
ml= MyClass()
print(ml.q)
```

- Tạo một lớp có tên Nhan_vien, sử dụng hàm **__init__()** để gán giá trị cho ten và tuoi:

```
class Nhan_vien:
  def __init__(self, ten, tuoi):
    self.ten = ten
    self.tuoi =tuoi
nv = Nhan_vien("Chi", 25)
print(nv.ten)
print(nv.tuoi)
```

II. Thực hành

Câu 1: Bài tập tham số hóa constructor

```
#Phép cộng hai số
class Addition:
    first = 0
    second = 0
    answer = 0

    # Hàm Khởi tạo
    def __init__(self, f, s):
        self.first = f
        self.second = s
    # Hàm hiển thị
    def view(self):
        print("Số thứ nhất = " + str(self.first))
        print("Số thứ hai = " + str(self.second))
        print("Phép cộng hai số= " + str(self.answer))
    # Hàm tính cộng
    def calculate(self):
        self.answer = self.first + self.second
# tạo đối tượng của lớp, tham số hóa hàm đượctạo
phepcong1 = Addition(200, 800)
# tạo đối tượng thứ hai của cùng một lớp
phepcong2 = Addition(15, 35)
# thực hiện Phép cộng trên đối tượng 1
phepcong1.calculate()
# thực hiện Phép cộng trên đối tượng 1
phepcong2.calculate()
# View kết quả của đối tượng 1
phepcong1.view()
# View kết quả của đối tượng 2
phepcong2.view()
```

Câu 2: Lớp Kế thừa_Xuất thông tin của Trưởng phòng và nhân viên trong Nhân sự

Trong đó:

Lớp Cha: Nhansu()

Lớp con Truongphong() kế thừa lớp Cha là Nhansu()

Lớp con Nhanvien() kế thừa lớp Cha là Truongphong()

```
# Tạo class Nhansu
class Nhansu():

    # __init__ được gọi là hàm tạo
    def __init__(self, name, idNum, salary, post):
        self.name = name
        self.idNum = idNum
        self.salary = salary
        self.post = post

    def view(self):
        print(self.name)
        print(self.idNum)
        print(self.salary)
        print(self.post)

# class kế thừa
class Truongphong(Nhansu):
    def __init__(self, name, idNum, salary, post):
        # gọi hàm __init__ của lớp cha
        Nhansu.__init__(self, name, idNum, salary, post)

# class kế thừa
class Nhanvien(Truongphong):
    def __init__(self):
        return()

# tạo một biến đối tượng gán tham số
tp = Truongphong('Chi', 111 , 90000 , 'Trưởng phòng')
ns = Nhansu('Anh', 112, 50000, "Nhân viên")

# gọi hàm của lớp Nhansu để view
tp.view()
ns.view()
```

----------000----------