Nama : Damianus Christopher Samosir

NPM : G1A022028

Kelas : Informatika B

M.Kuliah : Pemrograman Berbasis Objek

1. Jelaskan perbedaan pemrograman fungsional dan Pemrograman berorientasi Objek (OOP) pada Python beserta contoh pada program masing-masing !

Pemrograman fungsional dan pemrograman berorientasi objek (OOP) adalah dua paradigma pemrograman yang berbeda di Python. Pemrograman fungsional adalah paradigma pemrograman yang menekankan penggunaan fungsi untuk menyelesaikan masalah. Dalam pemrograman fungsional, fungsi diperlakukan sebagai warga negara kelas satu, yang berarti bahwa fungsi tersebut dapat diteruskan sebagai argumen ke fungsi lain, dikembalikan sebagai nilai dari fungsi, dan ditugaskan ke variabel. Pemrograman fungsional menghindari perubahan status dan data yang dapat diubah, dan sebagai gantinya berfokus pada data yang tidak dapat diubah dan fungsi murni yang tidak memiliki efek samping.

Di sisi lain, OOP adalah paradigma pemrograman yang menekankan penggunaan objek untuk menyelesaikan masalah. Dalam OOP, objek adalah turunan dari kelas, yang mengenkapsulasi data dan perilaku. OOP berfokus pada konsep pewarisan, polimorfisme, dan enkapsulasi. Warisan memungkinkan kelas untuk mewarisi properti dan metode dari kelas lain, polimorfisme memungkinkan objek mengambil berbagai bentuk, dan enkapsulasi memungkinkan data disembunyikan dari bagian lain dari program.Di Python, pemrograman fungsional dan OOP dapat digunakan untuk memecahkan masalah. Pemrograman fungsional dapat digunakan untuk menulis kode yang ringkas dan mudah dibaca yang mudah untuk diuji dan di-debug. OOP dapat digunakan untuk menulis kode modular dan dapat digunakan kembali yang mudah dipelihara dan diperluas.

Singkatnya, perbedaan utama antara pemrograman fungsional dan OOP dengan Python adalah pendekatan pemecahan masalah. Pemrograman fungsional menekankan penggunaan fungsi dan data yang tidak dapat diubah, sedangkan OOP menekankan penggunaan objek dan enkapsulasi.

Berikut contoh kode Programmng:

#kode Programming

def multiply\_by\_two(numbers):

    return list(map(lambda x: x \* 2, numbers))

numbers = [1, 2, 3, 4, 5]

result = multiply\_by\_two(numbers)

print(result)

Berikut contoh kode OOP:

#kode OOP

class Person:

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

        self.name = name

        self.age = age

    def say\_hello(self):

        print(f"Hello, my name is {self.name} and I'm {self.age} years old.")

person = Person("Chris", 19)

person.say\_hello()