**📌 Obyektga yo‘naltirilgan dasturlash (OOP) asoslari**

**🔹 OOP nima?**

OOP – bu **obyektga yo‘naltirilgan dasturlash** bo‘lib, dasturiy ta’minotni obyektlar yordamida tashkil qilish usuli. Har bir obyekt o‘zida **ma’lumotlar (atributlar)** va shu ma’lumotlarga ishlov beruvchi \*\*funksiyalar (metodlar)\*\*ni birlashtiradi.

Masalan, “Talaba” obyektida – ism, yosh, kurs kabi **atributlar**, o‘qish(), dam\_olish() kabi **metodlar** bo‘lishi mumkin.

**🔹 OOP asosiy tushunchalari**

1. **Klass (Class)**
   * Obyektlarni yaratish uchun **shablon**.
   * Unda atributlar va metodlar belgilanadi.
   * class kalit so‘zi yordamida e’lon qilinadi.
2. **Obyekt (Object / Instance)**
   * Klassdan yaratilgan aniq nusxa.
   * Har bir obyektning o‘ziga xos qiymatlari bo‘ladi.
   * Yaratish: obj = ClassName()
3. **Atributlar (Attributes)**
   * Klass yoki obyektning **o‘zgaruvchilari**.
   * Masalan: ism, yosh, rang, narx va hokazo.
4. **Metodlar (Methods)**
   * Klass ichida yoziladigan **funksiyalar**.
   * Obyektning xatti-harakatlarini belgilaydi.
   * Masalan: yurish(), gapirish(), hisoblash()
5. **Konstruktor (\_\_init\_\_ metodi)**
   * Klassdan obyekt yaratilganda avtomatik ishlaydigan metod.
   * Parametrlar qabul qilib, obyekt atributlarini boshlang‘ich qiymatlar bilan to‘ldiradi.
6. class Student:
7. def \_\_init\_\_(self, name, age):
8. self.name = name
9. self.age = age
10. **self kalit so‘zi**
    * Obyektning o‘ziga murojaat qilish uchun ishlatiladi.
    * self yordamida atributlar va metodlarga obyekt darajasida murojaat qilinadi.

**🔹 OOPning asosiy tamoyillari (4 ta poydevor)**

1. **Inkapsulyatsiya (Encapsulation)**
   * Ma’lumotlar va funksiyalarni **bir joyga jamlash**.
   * Ichki ma’lumotlarni tashqi dunyodan himoya qilish imkonini beradi.
2. **Abstraksiya (Abstraction)**
   * Keraksiz tafsilotlarni yashirish, faqat muhim jihatlarni ko‘rsatish.
   * Masalan, car.start() – bizga faqat mashina yurishi kerak, ichki dvigatel qanday ishlashini bilish shart emas.
3. **Merosxo‘rlik (Inheritance)**
   * Bitta klassdan yangi klass olish.
   * Yangi klass ota klassning atributlari va metodlarini meros qilib oladi.
4. **Polimorfizm (Polymorphism)**
   * Bir xil nomli metodlarning turli klasslarda har xil ishlashi.
   * Masalan, draw() metodi Circle klassida aylana chizsa, Rectangle klassida to‘rtburchak chizadi.

Shunday qilib, OOP orqali kod **modulli, qayta foydalanish mumkin bo‘lgan va tushunarli** bo‘ladi.

❓ Xohlaysizmi, men shu asosiy tushunchalarni ham misollar va mashqlar bilan ko‘rsatib beray?