AN

AN SOLUTION PAKISTAN

Short Questions & Answers

Computer Science

Chapter 3 — Object Oriented Programming Using Python

Easy English answers · with Urdu meaning (matlab) for each question

A Basics of OOP

Q1. What is Object-Oriented Programming (OOP)?

OOP is a way of making programs using **objects** and **classes**. Each object is like a real-world thing that has data and can do actions.

> **Matlab:** OOP programming ka aisa tareeqa hai jisme hum objects aur classes bana kar program likhte hain. Har object asli cheez ki tarah hota hai jiske paas data bhi hota hai aur kaam bhi karta hai.

Q2. What is a class?

A class is a **plan (blueprint)**. It tells what data and actions an object will have.

> **Matlab:** Class ek naksha/design hai jo batata hai ke object ke paas kaunsa data aur kaunse kaam honge.

Q3. What is an object?

An object is a **real thing made from a class**. It has real values.

> **Matlab:** Object class se banee asli cheez hai jiske paas asli values hoti hain.

Q4. What is the relation between a class and an object?

A class is like a **plan**, and an object is the **real thing made** from that plan. One class can make many objects.

> **Matlab:** Class naksha hai, object us nakshe se banee cheez. Ek class se bohot saare objects ban sakte hain.

Q5. What are attributes and methods?

Attributes are the data (like `name, age`). **Methods** are the actions (like `eat()`, `sleep()`).

> **Matlab:** Attributes = data (naam, umar). Methods = kaam (khana, sona).

Q6. Write any two benefits of OOP.

(1) **Modularity** — code is divided into small parts. (2) **Reusability** — the same code can be used again.

> **Matlab:** Do faide — Modularity (code chhote hisson mein bat jata hai) aur Reusability (wahi code dobara use ho jata hai).

Q7. Why is OOP easy to maintain?

Because changing one part does **not break the whole program**.

> **Matlab:** Kyunki ek hissa badalne se poora program kharaab nahi hota.

Q8. Name the core concepts of OOP.

Classes and objects, **encapsulation**, **inheritance**, and **polymorphism**.

> **Matlab:** Classes/objects, encapsulation, inheritance, aur polymorphism.

Q9. What is the use of the `__init__` method?

It **sets the starting values** (like name) when a new object is made.

> **Matlab:** Jab naya object banta hai to `__init__` shuru ki values (jaise naam) set kar deta hai.

Q10. What does `self` mean?

`self` means the **current object**. Python sends it to methods by itself.

> **Matlab:** `self` ka matlab "abhi wala object". Python khud ise methods ko bhej deta hai.

Q11. How do you make an object?

Write the class name with values, like `object1 = Student("Ali")`.

> **Matlab:** Class ka naam values ke saath likho, jaise `object1 = Student("Ali")`.

B Encapsulation & Access Control

Q12. What is encapsulation?

Encapsulation means putting **data and methods together in one class**, and **hiding the data** from direct use.

> **Matlab:** Data aur methods ko ek class mein rakhna, aur data ko direct use se chhupa dena.

Q13. Does Python have strict "private" or "protected" keywords?

No. Unlike Java or C++, Python has no such strict words. It uses **naming rules (underscores)** instead.

> **Matlab:** Nahi. Java/C++ ki tarah Python mein ye keywords nahi hote; ye underscore wale naam use karta hai.

Q14. Why is encapsulation important?

It **keeps data safe**, gives security, makes code easy to manage, and allows controlled change.

> **Matlab:** Ye data ko mehfooz rakhta hai, security deta hai, aur code manage karna aasan karta hai.

Q15. Give a real-life example of encapsulation.

A **bank account** — the balance is hidden. You can only use `deposit()` and `withdraw()`. You cannot change the balance directly.

> **Matlab:** Bank account — balance chhupa hota hai, sirf deposit/withdraw se chalta hai; direct nahi badal sakte.

Q16. What are the levels of access control in Python?

Public, **Protected**, and **Private**.

> **Matlab:** Teen levels — Public, Protected, Private.

Q17. What are public members?

They can be used from **anywhere** (inside or outside the class). In Python, members are public by default. Written as `self.name`.

> **Matlab:** Public members kahin se bhi use ho sakte hain; Python mein by default public hote hain.

Q18. What are protected members?

They are for use **inside the class and its child classes**. It is only a sign, not a strict rule. Written as `self._name` (one underscore).

> **Matlab:** Protected class aur uski child class ke andar ke liye; sirf ishaara hai. `self._name` (ek underscore).

Q19. What are private members?

They **cannot be used easily from outside** the class. Written as `self.__name` (two underscores).

> **Matlab:** Private members bahar se aasani se use nahi hote. `self.__name` (do underscore).

Q20. What is name mangling?

It is Python's way of **changing a private variable's name inside**, so it is not used by mistake.

> **Matlab:** Name mangling mein Python private variable ka naam andar-hi-andar badal deta hai taake galti se use na ho.

Q21. In the BankAccount example, what is the output of `print(account.get_balance())` ?

1200 (because 1000 + 500 − 300 = 1200).

> **Matlab:** 1200 — pehle 1000, 500 jama (deposit), 300 nikaale (withdraw). 1000 + 500 − 300 = 1200.

Q22. Why does `print(account.__balance)` give an error?

Because `__balance` is **private**, so we cannot use it directly from outside the class.

> **Matlab:** `__balance` private hai, is liye bahar se direct use nahi hota — error aata hai.

Easy trick: no underscore `name` = public · one underscore `_name` = protected · two underscores `__name` = private.

C Inheritance

Q23. What is inheritance?

Inheritance means one class **takes the data and methods of another class**. The new (child) class can add or change things.

> **Matlab:** Ek class dusri class ka data aur methods le leti hai. Child class usme naya add ya change kar sakti hai.

Q24. What are the benefits of inheritance?

It **reuses code**, stops repeating, and makes it easy to add more without writing again.

> **Matlab:** Code dobara use hota hai, repeat nahi karna padta, aur aage barhana aasan hota hai.

Q25. What is single inheritance?

When a child class takes from **only one parent class**.

> **Matlab:** Jab child sirf ek hi parent class se cheezein leti hai.

Q26. What is a parent class (base class)?

The class **whose things are taken**. Example: `Animal`.

> **Matlab:** Wo class jiski cheezein li jaati hain. (misaal: Animal)

Q27. What is a child class (derived class)?

The class that **takes from the parent**. Written as `class Child(Parent):`

> **Matlab:** Wo class jo parent se cheezein leti hai. `class Child(Parent):`

Q28. What is method overriding?

When a child class writes its **own new version** of a method that the parent already has.

> **Matlab:** Child wahi method apne tareeqe se dobara (naya) likh deti hai.

Q29. What is the use of `super()` ?

`super()` is used to **call the parent class's method**.

> **Matlab:** `super()` parent class ka method bulaane ke liye use hota hai.

Q30. What is multiple inheritance?

When one child class takes from **more than one parent class**. Example: a Smartphone from Camera and Computer.

> **Matlab:** Jab ek child ek se zyada parents se cheezein leti hai (jaise Smartphone — Camera + Computer se).

Q31. What is multilevel inheritance?

A chain: **Grandparent → Parent → Child**. The child can use things from both parent and grandparent.

> **Matlab:** Ek chain (Grandparent → Parent → Child). Child ko parent aur grandparent dono ki cheezein milti hain.

Q32. In multilevel inheritance, what can the grandchild class use?

All the **methods and data** of its parent and its grandparent.

> **Matlab:** Apne parent aur grandparent — dono ki saari cheezein.

Q33. What is a subclass?

A subclass is another name for the **child class** — the class that takes from a parent.

> **Matlab:** Subclass = child class ka dusra naam (jo parent se leti hai).

D Example / Code Questions

Q34. What does the `display()` method do in the Student class?

It **prints the student's name**.

> **Matlab:** Ye student ka naam print (dikha) deta hai.

Q35. In `class Dog(Animal):` , which inheritance is this?

Single inheritance. Dog is the child, Animal is the parent.

> **Matlab:** Single inheritance — Dog child hai, Animal parent.

Q36. If `my_dog = Dog("Buddy")` , what is the output of `my_dog.eat()` ?

`Buddy is eating.` (`eat()` came from the parent class Animal.)

> **Matlab:** "Buddy is eating." — `eat()` parent (Animal) se mila hai.

Q37. What is the output of `my_dog.bark()` ?

`Buddy says Woof!` (`bark()` is in the child class Dog.)

> **Matlab:** "Buddy says Woof!" — `bark()` child (Dog) ka apna method hai.

Q38. In `class Child(Father, Mother):` , which inheritance is this?

Multiple inheritance (two parents).

> **Matlab:** Multiple inheritance — do parents (Father aur Mother).

Q39. Which sign is used for a private variable?

Two underscores `__` , like `self.__balance` .

> **Matlab:** Do underscore __, jaise `self.__balance`.

Q40. Which sign is used for a protected member?

One underscore `_` , like `self._name` .

> **Matlab:** Ek underscore _, jaise `self._name`.

Q41. Write a simple Python class with one attribute and one method.

A basic example from the chapter:

```
class Student:
    def __init__(self, name):
        self.name = name

    def display(self):
        print("Student Name:", self.name)
```

```
object1 = Student("Ali")
object1.display()      # Student Name: Ali
```

> **Matlab:** `Student` ek class hai, `name` uska attribute, aur `display()` uska method jo naam print karta hai.

Test ke liye tip: short definitions yaad karein — OOP, class, object, encapsulation, inheritance aur uski 3 types — aur 3 access levels (public / _ protected / __ private). Yehi sabse zyada aate hain.