

Instance Method Questions

1. What is an instance method in Python? (Explain with a simple example.)

An instance method is a function defined inside a class that operates on an instance of the class.

```
class welcome:
```

```
    def hello(self):  
        print("Hello!")
```

```
p = welcome()
```

```
p.hello()
```

2. Write a class `Car` with an instance method `start_engine` that prints "Engine started."

```
class Car:
```

```
    def start_engine(self):  
        print("Engine started.")
```

```
my_car = Car()
```

```
my_car.start_engine()
```

3. How do you call an instance method — with the class name or with the object? (Give example.)

Instance methods are called using the object of the class.

```
class Color:
```

```
    def rainbow(self):  
        print("VIBGYOR")
```

```
p = Color()
```

```
p.rainbow()
```

4. Create a class `Circle` that takes radius in `__init__` and has an instance method `area()` to calculate the area of the circle.

```
class Circle:
```

```
    def __init__(self, radius):
```

```
        self.radius = radius
```

```
    def area(self):
```

```
        return 3.14 * self.radius ** 2
```

```
c = Circle(5)
```

```
print(c.area())
```

5. What will happen if you try to call an instance method without creating an object?

We will get a TypeError.

Class Method Questions

6. What is a class method? How is it different from an instance method? (Answer with example.)

A class method is a method that takes cls as the first parameter and operates on the class, not the instance. It is defined using the `@classmethod` decorator.

- Instance methods use self and work with object data.

- Class methods use cls and work with class-level data.

```
class Colors:
```

```
    name = "Magenta"
```

```
    @classmethod
```

```
    def show_name(cls):
```

```
        print(cls.name)
```

```
Colors.show_name()
```

7. Write a class `Employee` with a class variable `company = "TechCorp"` and a class method `get_company_name()`.

```
class Employee:

    company = "TechCorp"

    @classmethod

    def get_company_name(cls):

        return cls.company

print(Employee.get_company_name())
```

8. How do you define a class method? (Which decorator is used?)

We define it with the `@classmethod` decorator before the method.

```
class Welcome:

    @classmethod

    def sample(cls):

        print("Welcome! This is a class method.")
```

9. Create a class `Counter` with a class variable `count` and a class method `increment()` that increases the count by 1.

```
class Counter:

    count = 0

    @classmethod

    def increment(cls):

        cls.count += 1

Counter.increment()

print(Counter.count)
```

Static Method Questions

10. What is a static method in Python? (Explain with example.)

A static method is like a normal function placed inside a class. It doesn't use self or cls, so it doesn't depend on object or class data.

```
class Welcome:
```

```
    @staticmethod
```

```
    def hello():
```

```
        print("Hello!")
```

```
Welcome.hello()
```

11. Write a static method `is_even(num)` inside a class `MathUtils` that returns `True` if the number is even, else `False`.

```
class MathUtils:
```

```
    @staticmethod
```

```
    def is_even(num):
```

```
        return num % 2 == 0
```

```
print(MathUtils.is_even(24))
```

```
print(MathUtils.is_even(25))
```

12. Which method type should you use when:

- You want to access/modify class variables — (Instance/Class/Static)?

Class Method

- You don't need class or instance data — (Instance/Class/Static)?

Static Method