

Topics Of the Day4

- INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING
- WHAT IS A CONSTRUCTOR
- GETTERS AND SETTERS IN PYTHON
- INHERITANCE IN PYTHON
- TYPES OF ACCESS MODIFIERS IN PYTHON
- INSTANCE VARIABLES VS CLASS VARIABLES
- Single Level Inheritance in Python
- What is Multiple Inheritance?
- What is Multilevel Inheritance:
- What is Hybrid Inheritance
- Super()Keyword in Python:

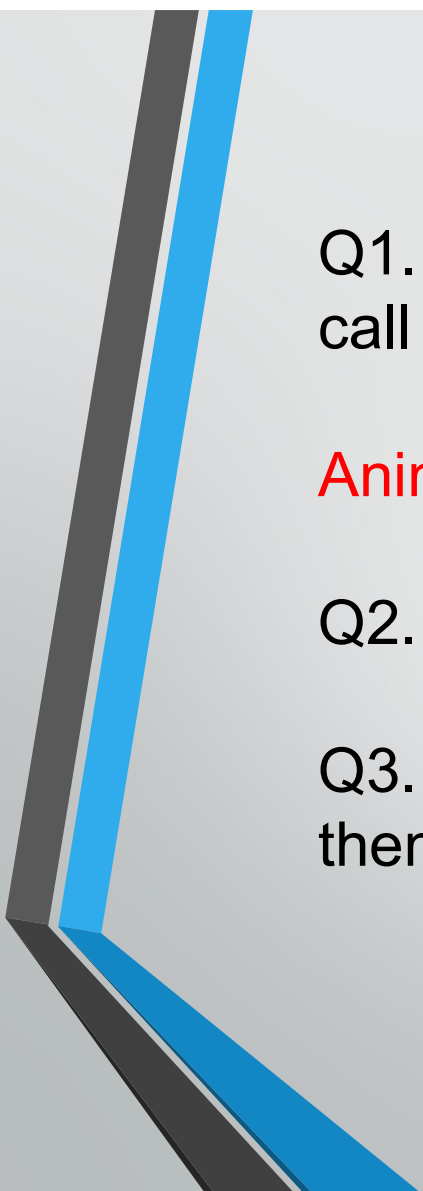
Q1. Create a class Circle with radius and a method to calculate area.

```
Class Circle:  
    pass
```

Q2. Create a class Employee with a constructor that stores name and salary.

```
Class Employee :  
    pass
```

Q3. Create a setter method to update age only if it is ≥ 18 .

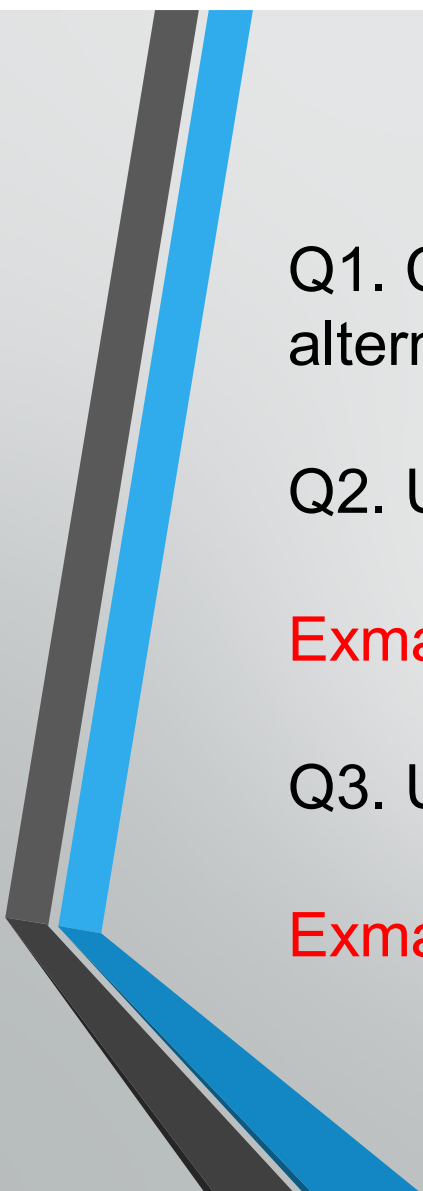


Q1. Create Animal → Dog and create given method then call dog's method.

Animal → info Dog → bark

Q2. Access a private variable (id) using name-mangling

Q3. Create **instance variables** for name and age and show them for two objects



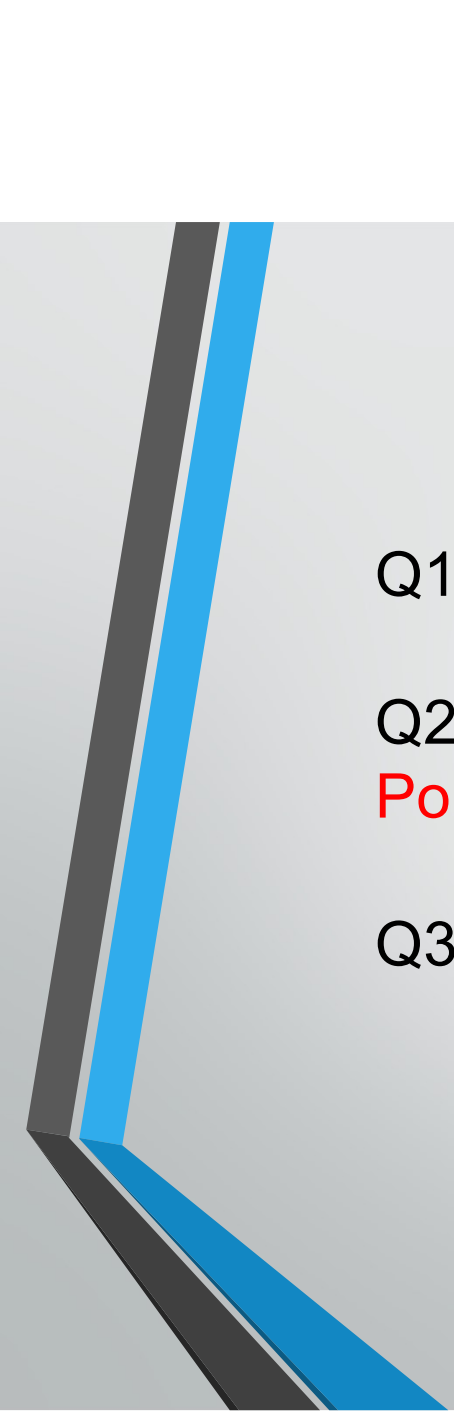
Q1. Create object from a string "name-age-id" using alternative constructor

Q2. Use `super()` to call parent constructor.

Exmaple: `Base(name,age) → Derived(name,age,language)`

Q3. Use `super()` to access parent class method.

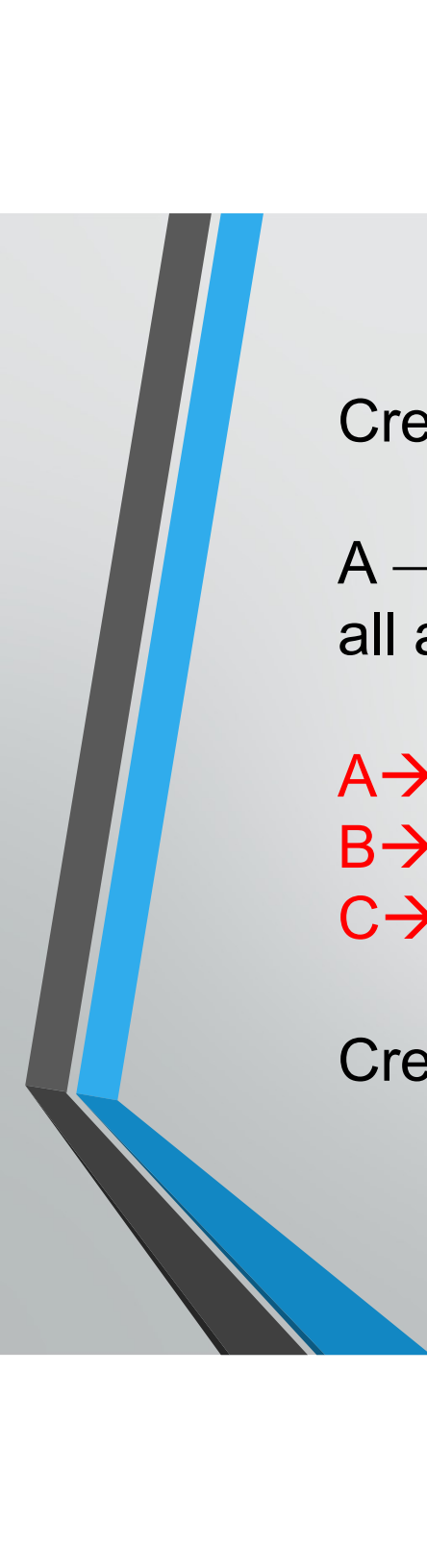
Exmaple: `Base → showinfo()` and `Derived showinfo()`



Q1. Create a class using `__str__()` method.

Q2. Overload `*` operator for multiplying object values.
`Point(4,6) * Point(2,5) → Point(8,30)`

Q3. Add new method in child not in parent



Create class A and B, and class C inheriting from A and B.

A \rightarrow B \rightarrow C (three-level inheritance) and call methods of all and use `super()` to pass value

A \rightarrow name

B \rightarrow age

C \rightarrow course

Create a hybrid structure: A \rightarrow B, A \rightarrow C, C \rightarrow D.