

# OOP day 1 | Need to know topics

## 1. Class and `__init__`

**Question 1:** Create a **Book** class with title, author, and year of publication.

- When creating an object, these 3 values should be initialized using the `__init__` method.
- Then, print the book details using a method.

**Hint:**

- Use `self.title`, `self.author`, `self.year` in the `__init__`.
  - Create a method `show_details(self)` to print details.
- 

## 2. `__init__` with Default Arguments

**Question 2:** Create a class **Student** with name and grade. If no grade is given, use "Not Assigned" as default.

**Hint:**

- Define `def __init__(self, name, grade="Not Assigned")`.
- 

## 3. Simple Methods

**Question 3:** Create a class **Calculator** that takes two numbers and provides methods:

- `add(self)`
- `subtract(self)`
- `multiply(self)`

- `divide(self)`

**Hint:**

- Store numbers in `__init__`, and create a method for each operation using `self.num1` and `self.num2`.
- 

## 4. Instance Attribute vs Class Attribute

**Question 4: Create a class `Dog`:**

- Every dog has a name (instance attribute).
- All dogs belong to species "`Canine`" (class attribute).

**Hint:**

- Define `species = "Canine"` outside `__init__`.
  - Define `self.name = name` inside `__init__`.
  - Try printing `dog1.species` and `Dog.species`.
- 

## 5. Instance Method vs Class Method

**Question 5: Create a class `School`:**

- Instance attribute: `student_name`
- Class attribute: `school_name = "ABC Public School"`
- Instance method: `get_student_info(self)` → prints student name and school
- Class method: `change_school_name(cls, new_name)` → changes school name

**Hint:**

- Use `@classmethod` decorator for class method.
  - Use `cls.school_name = new_name` inside class method.
- 

## Bonus Real-Life Scenario Based (Combining above)

### Question 6: Create a class `BankAccount`:

- On creation, accept `name` and `balance` (default: 0).
- Method: `deposit(self, amount)`
- Method: `withdraw(self, amount)`
- Method: `display_balance(self)`

#### Hint:

- Initialize `balance` to 0 if not given.
- In `deposit`, add amount to balance.
- In `withdraw`, subtract only if enough balance exists.
- Use `self.balance`.