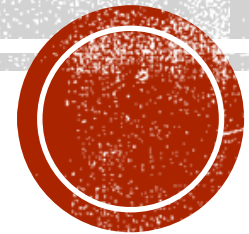


# CLASSES

Day 2 NTI Sohag



# TASK1

## ■ Problem Statement:

- Create a base class Vehicle with attributes make, model, and year.
- Define a method vehicle\_info that returns a string with these details.
- Then, create two derived classes: Car and Motorcycle.
- The Car class should have an additional attribute number\_of\_doors, and the Motorcycle class should have an additional attribute type\_of\_handlebars.
- Override the vehicle\_info method in both derived classes to include these additional details.

## ■ Requirements:

- Vehicle Class: Create the base class with common attributes make, model, and year.
- Derived Classes: Create Car and Motorcycle classes inheriting from Vehicle.
- Method Overriding: Override the vehicle\_info method in the derived classes to include specific details.
- Demonstration: Instantiate objects of Car and Motorcycle, and print their information using the vehicle\_info method.



# TASK2

- **Problem Statement:**

- Given a list of integers, write a function `find_squares` that returns a list of the squares of all even numbers in the original list using list comprehension.
- If the list is empty, return an empty list.

- **Requirements:**

- List Comprehension: Use list comprehension to filter even numbers and compute their squares.
- Function Implementation: Implement the `find_squares` function.
- Edge Case Handling: Ensure the function handles an empty list correctly.
- Demonstration: Test the function with various lists, including lists with odd numbers, even numbers, and an empty list.



# TASK3

- **Problem Statement:**

- Write a function called `order_summary` that takes two types of arguments:
  - Any number of positional arguments (`*args`) representing the names of items in an order.
  - Any number of keyword arguments (`**kwargs`) where the keys are item names and the values are the quantities of those items.
  - The function should print a summary of the order, showing each item and its corresponding quantity. If an item is listed in `*args` but not in `**kwargs`, assume the quantity for that item is 1.

- **Requirements:**

- Use of `*args`: Handle multiple item names as positional arguments.
- Use of `**kwargs`: Handle item quantities as keyword arguments.
- Logic: Combine `*args` and `**kwargs` to print an order summary, setting the quantity to 1 for items that do not have a specified quantity in `**kwargs`.
- Example Demonstration: Show how the function works with various input combinations, including cases where some items do not have specified quantities.

