

Assignment: Introduction to OOP in Python

Objective: The goal of this assignment is to practice the foundational concepts of Object-Oriented Programming (OOP) learned in Class 1 by creating and working with Python classes and objects.

Task 1: Create Your First Class

1. Create a class named `Person` with the following attributes:
 - `name` (string): The name of the person.
 - `age` (integer): The age of the person.
 2. Add the following methods to the class:
 - `__init__`: Initializes the `name` and `age` attributes when a `Person` object is created.
 - `greet`: Prints a greeting message in the format: "Hello, my name is <name> and I am <age> years old."
 3. Create at least two objects of the `Person` class and call the `greet` method for each object.
-

Task 2: Modify Attributes

1. Add a method `birthday` to the `Person` class that:
 - Increases the `age` of the person by 1.
 - Prints a message: "Happy Birthday <name>! You are now <age> years old."
 2. Demonstrate the `birthday` method by calling it on one of the `Person` objects created earlier.
-

Task 3: Practice with Multiple Methods

1. Add a new attribute `hobby` (string) to the `Person` class and modify the `__init__` method to accept it.
 2. Add a method `share_hobby` that:
 - Prints a message: "Hi, I am <name> and I love <hobby>."
 3. Create a new `Person` object with a hobby and call the `share_hobby` method.
-

Submission Guidelines

- Save your Python script file named `Class1_OOP_Assignment_<YourName>.py`.
 - Submit your assignment by screenshots.
 - Make sure to show both the inputs and outputs.
 - Include comments in your code to explain each method and attribute.
-

Bonus Task: Create a class `Animal` with attributes like `species` and `sound`. Add a method `make_sound` that prints the sound the animal makes. Create objects for different animals and call their `make_sound` methods.