**Name:** Hamsa M Ahmed
**Subject:** OOP Practical
**Department:** Computer Network System

# Lecture 2: Classes and Objects in Depth

## 1. Introduction

In the previous lecture, we learned the basics of OOP and how classes and objects form its core. In this lecture, we will dive deeper into how to create classes, instantiate objects, and use class attributes and methods effectively.

## 2. Defining a Class in Python

A class is defined using the `class` keyword followed by the class name and a colon.

```
class Person:
```

This creates an empty class named `Person`.

## 3. Attributes and Methods

Classes can have attributes (variables) and methods (functions).

```
class Person:
    def __init__(self, name, age):
        self.name = name  # attribute
        self.age = age

    def greet(self):
        print(f"Hello, my name is {self.name} and I am {self.age} years
old.")
```

## 4. Creating Objects

Objects are created by calling the class as if it were a function.

```
person1 = Person("Alice", 30)
person1.greet()
```

**Name:** Hamsa M Ahmed
**Subject:** OOP Practical
**Department:** Computer Network System

Output:

```
Hello, my name is Alice and I am 30 years old.
```

# 5. Class vs Instance Variables

- **Instance variables** are unique to each object (e.g., `self.name`, `self.age`).
- **Class variables** are shared among all instances of the class.

```
class Dog:
    species = "Canis familiaris"  # class variable

    def __init__(self, name):
        self.name = name  # instance variable

dog1 = Dog("Buddy")
dog2 = Dog("Max")
print(dog1.species)
print(dog2.species)
```

Output:

```
Canis familiaris
Canis familiaris
```

# 6. Example: Managing Class and Instance Variables

```
class Employee:
    raise_amount = 1.05  # class variable

    def __init__(self, first, last, salary):
        self.first = first
        self.last = last
        self.salary = salary

    def apply_raise(self):
        self.salary = int(self.salary * self.raise_amount)

emp1 = Employee("John", "Doe", 50000)
emp1.apply_raise()
print(emp1.salary)
```

**Name:** Hamsa M Ahmed
**Subject:** OOP Practical
**Department:** Computer Network System

# 7. Exercises

1. Create a `Car` class with instance variables for make, model, and year. Add a class variable for the total number of cars created.
2. Add methods to update the car's details and display them.
3. Instantiate multiple `Car` objects and print the total number of cars created.

# 8. Summary

This lecture covered in-depth the syntax of classes, the difference between class and instance variables, and how to use methods to manipulate object data.

# 9. References

1. Python Classes and Objects - Official Docs
2. W3Schools Python Classes