Learning Consolidation

# Introduction to Object-Oriented Programming

# Learning Objectives

- Explore objects in real life.
- Identify the noun and verbs given in a problem statement.
- Declare attributes as variables in a class.
- Define behavior as methods in a class.
- Explore OOP vs. procedural programming.

# What Is an Object?

An object is something material that may be perceived by the senses.

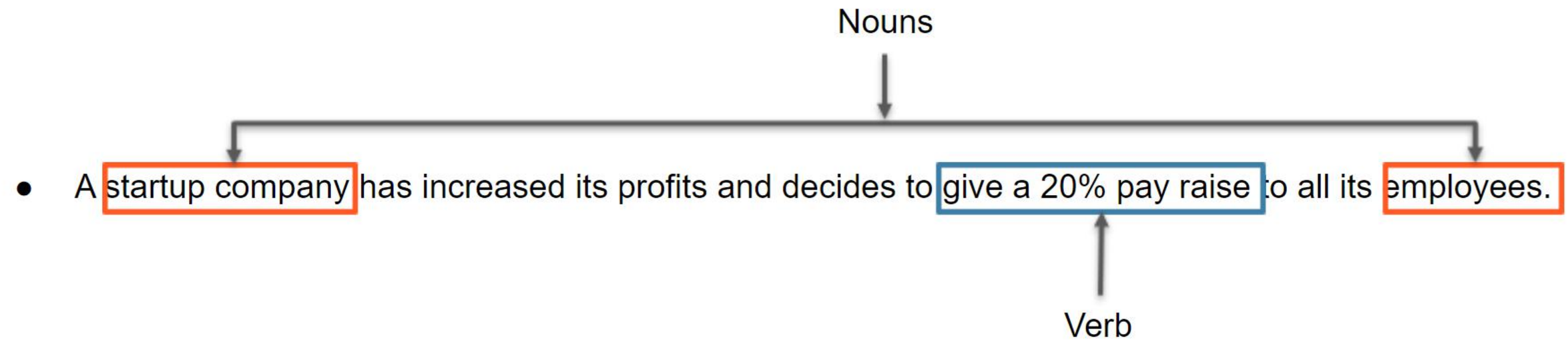
- Real-world objects share two characteristics: They all have properties and behavior.
- Consider a bicycle, which is a real-world object.
- The properties and behaviors of the bicycle are:

| Properties | Behavior |
|---|---|
| Color : Blue | Change gear |
| No. of gears: 7 | Maintain current speed |
| Is the disk brake enabled? Yes | Change pedal cadence |

# Bank Account as an Object

- Objects need not be only tangible.
- A bank account is not a tangible object, yet when you perform a banking transaction, it is necessary.
- Thus, a bank account can be considered an object in the real world.
- When you develop a banking application, the real-world object (i.e., bank account), can be modeled as an object in a program.

# Identifying Nouns and Verbs



# Attributes in a Class

- The class name is Employee.
- Attributes are modeled as variables in the class.

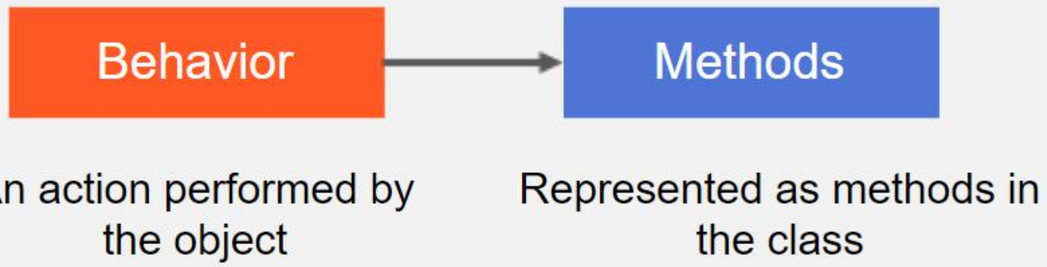

Uniquely identify an object

Represented as variables in the class

```
public class Employee {
    String employeeName;
    int employeeCode;
    int age;
    String dob;
    double salary;
```

# Behavior in a Class

- Behaviors are modeled as methods in the class.




```
double calculateSalaryHike(float hikePercentage){
    return employee.salary + (employee.salary * hikePercentage/100);
    }
```

# OOP vs. Procedural

| Procedural | OOP |
| --- | --- |
| It is process-oriented. | It is result-oriented. |
| It follows a top-down approach. | It follows a bottom-up approach. |
| Each function contains different data. | Each object controls its data. |
| There is no access specifier. | It has access specifiers, such as Private, Public, Protected, etc. |
| Adding new data and functions is not easy. | Adding new data and functions is easy. |
| Overloading is not possible. | Overloading is possible. |
| It is less secure. | It is more secure. |
| There is no concept of inheritance. | Inheritance is allowed. |