Inheritance

1. What is Inheritance?
   * Inheritance is the passing on of genetic traits from parents to their offspring, and these offspring get all the genetic information from their parents.
2. Why we use inheritance?
   * This feature promotes code reusability and creates a parent-child relationship between classes.
3. Terms Used in inheritance
   * Class: A class is a collection of objects with similar attributes. It's a blueprint or template from which items are made.
   * Sub Class/Child Class: Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
   * Super Class/Parent Class: The features of a subclass are inherited from the superclass. It's also known as a parent class or a base class.
   * Reusability: Reusability, as the name implies, is a feature that allows you to reuse the fields and methods of an existing class while creating a new one. The fields and methods defined in the preceding class can be reused.
4. Child constructor must call the super constructor

*class* Person {

  name: *string*;

  address: *string*;

***constructor***(*name*: *string*, *address*: *string*) {

    this.name = *name*;

    this.address = *address*;

  }

}  
*class* Employee extends Person {

  salary: *number*;

  ***constructor***(*name*: *string*, *address*: *string*, *salary*: *number*) {

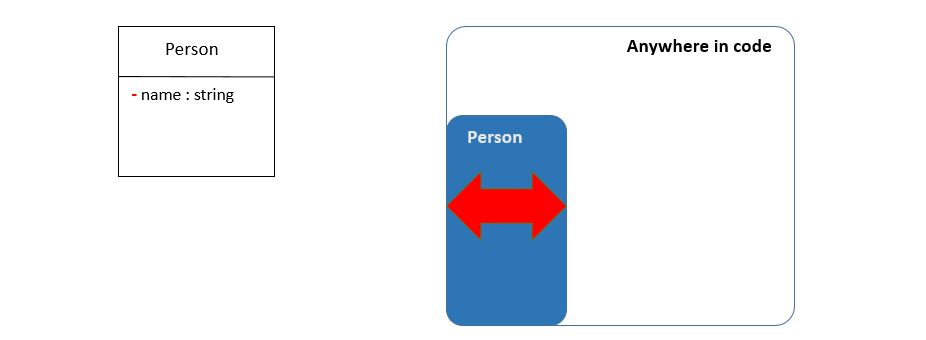
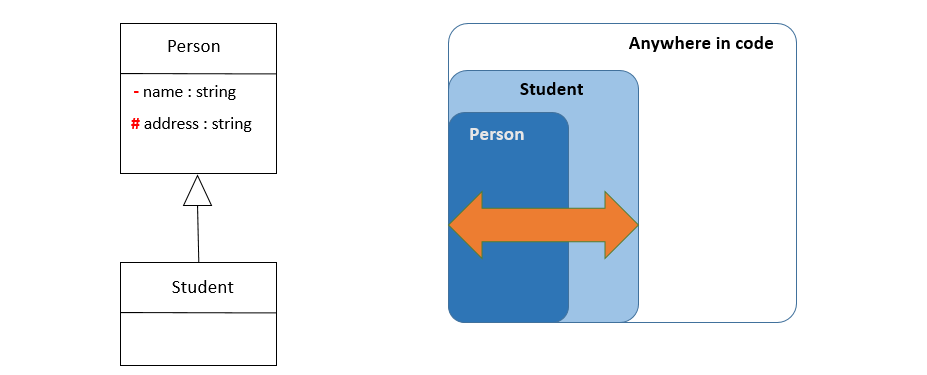
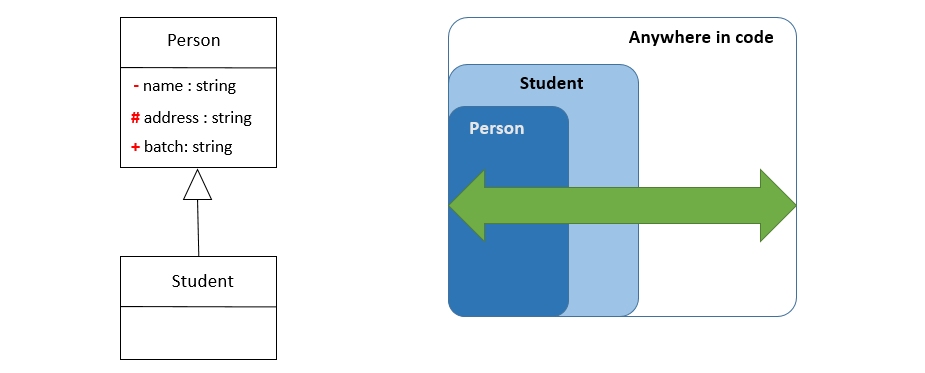
**super**(*name*, *address*);

    this.salary = *salary*;

}

}  
*let* ronan = new Employee("ronan", "paris", 400);

console.log(ronan);

1. Super
   * Super is a call to the super’s constructor.
   * The 'super' keyword allows referencing the parent class or superclass of a subclass.
2. Visibility: PRIVATE > PROTECTED > PUBLIC
   * PRIVATE attributes are visible only inside the CLASS
   * PROTECTED attributes are visible inside CLASS but also child CLASSES
   * PUBLIC attributes can be used from anywhere