

# Herencia en C++ (Parte 1)

# Definición del concepto (1)

Una de las características mas importantes del Paradigma Orientado a Objetos (POO) es la Herencia, la cual es una forma de reutilización de código, que permite implementar nuevas clases a partir de otras ya existentes, haciendo que las nuevas clases ***hijas*** adquieran los atributos y métodos definidos en la clase ***padre***.

En este sentido, en Herencia se usan como sinónimos los términos:

- Clase base y Clase derivada, o
- Clase padre y Clase hija, o
- SuperClase y SubClase

# Definición del concepto (2)

Al implementar la Herencia se va desde lo general o abstracto, hasta lo particular o definido. Por ejemplo, se pensamos en la entidad **Trabajador**, es algo muy amplio o abstracto, porque se pueden clasificar en categorías o tipos, es decir **Trabajador** no esta haciendo referencia a un tipo de trabajador en particular.

Por el contrario: Taxista, Operario de bascula, o Servicios Generales, si hacen referencia a un tipo de trabajador especifico.

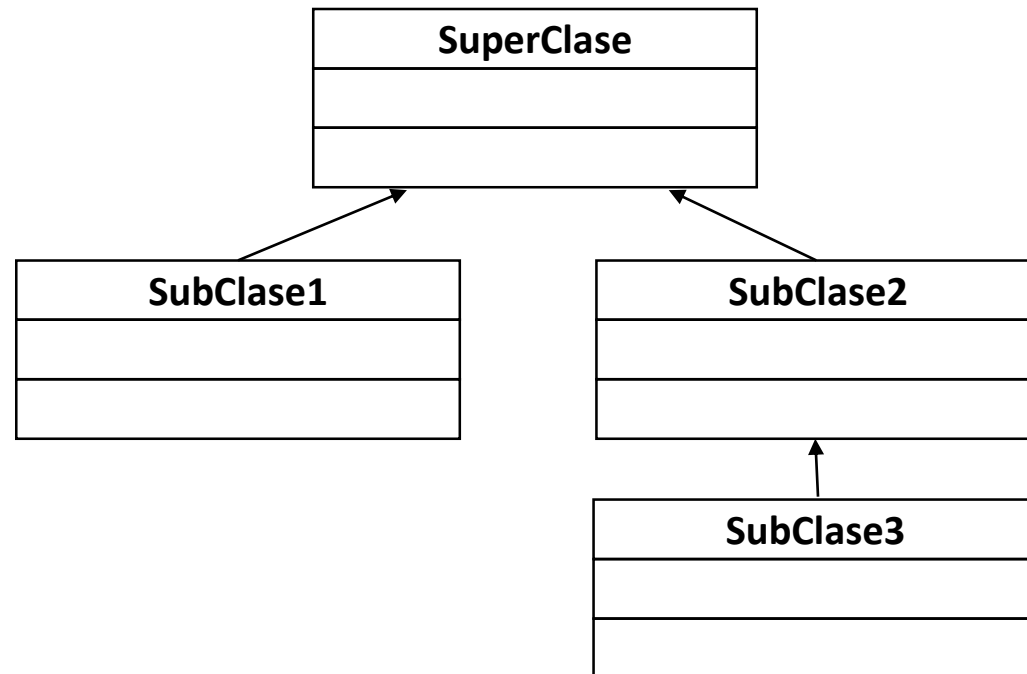
Es por ello que en la Herencia se usa el concepto de “**es un/una**”, o “**es un tipo de**”, es decir;

- Servicio Generales “**es un**” **Trabajador**, o
- Servicios Generales “**es un tipo de**” **Trabajador**

# Tipos de Herencia (1)

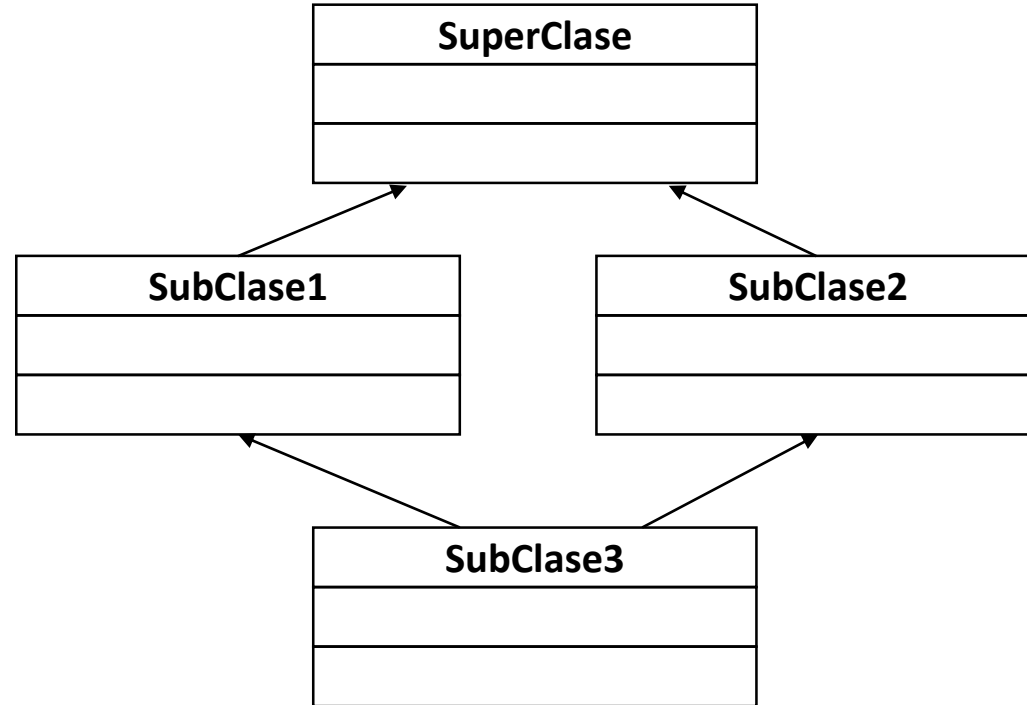
De forma básica C++ implementa dos tipos de Herencia; la Herencia Simple y la Múltiple.

En la Simple, una clase ***derivada*** hereda de una única clase ***base***.



# Tipos de Herencia (2)

En la Múltiple, una clase ***derivada*** hereda de dos clases ***bases***.



# Sintaxis

Para implementar la Herencia en C++ las **Clases hijas** deben especificar en la definición de la clase el nombre de la **Clase padre(s)** de la cual van a heredar.

La sintaxis es la siguiente:

```
class NombreClaseHija: acceso NombreClasePadre1, acceso NombreClasePadre2 {  
    public:  
        Declaración de atributos, constructores y métodos  
    private:  
        Declaración de atributos, constructores y métodos  
    protected:  
        Declaración de atributos, constructores y métodos  
};
```

# Ejemplo de Herencia (1)

Se requiere implementar una aplicación en C++ que gestione los datos y calcule el valor a pagar a los docentes de una institución universitaria, si se conoce que están categorizados en: Tiempo completo (DTC) y Hora catedra (DHC).

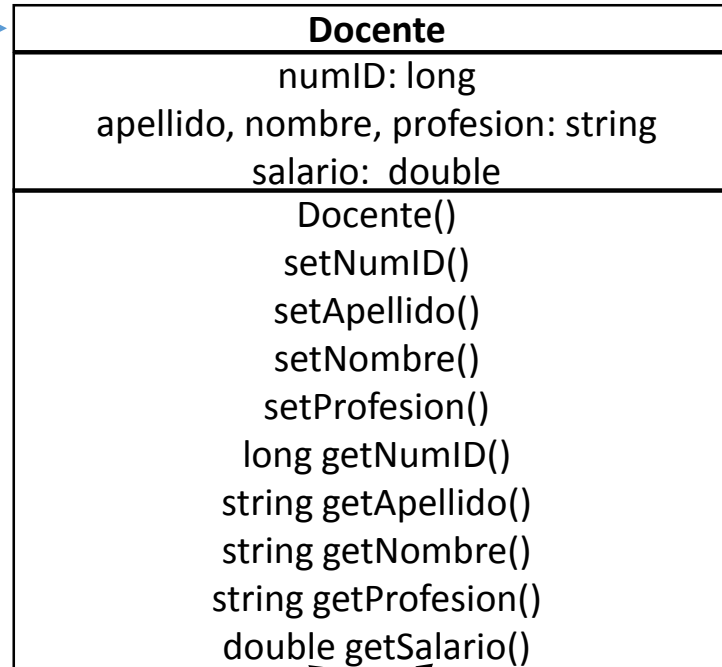
- Los DTC tienen un salario establecido del doble del SMMLV y una cantidad de horas mensuales mínimas de investigación (8 horas).
- Para calcular el pago de los DHC se tiene en cuenta la cantidad de horas trabajadas en el mes y el nivel de estudios.

En general el DTC y el DHC están descritos por: numero de identificación, apellido, nombre y nivel de estudios.

En este caso el DTC y el DHC (***Clases Hijas***) son un tipo de Docente (***Clase Padre***).

# Ejemplo de Herencia (2)

SuperClase



## Diagrama de Clases

Jerarquía de Clases de  
2 Niveles

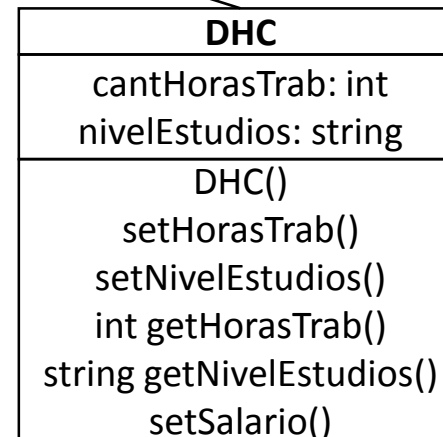
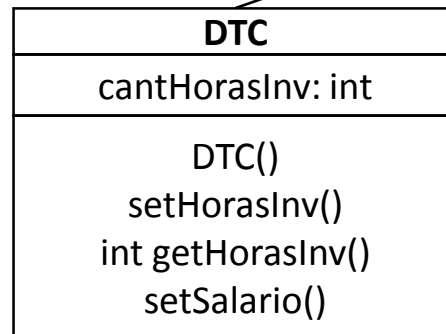
Nivel 1

*es un*

*es un*

Nivel 2

SubClases



**DTC** y **DHC** heredan de **Docente**, por lo tanto están adquiriendo las características (atributos y métodos) de **Docente**, es decir, **DTC** y **DHC** tienen numID, apellido, ..., pero **DTC** y **DHC** también están definiendo sus propias características



# Ejemplo de Herencia (2)

## Declaración de las Clases (.h)

### SuperClase

```
class Docente{
public:
    long numID;
    string apellido, nombre, profesion;
    double salario;

    Docente();
    virtual ~Docente();
    void setNumID();
    void setApellido();
    void setNombre();
    void setProfesion();
    long getNumID();
    string getApellido();
    string getNombre();
    string getProfesion();
    double getSalario();

protected:
private:
};
```

### SubClasses

```
class DTC: public Docente
{
public:
    int cantHorasInv;

    DTC();
    virtual ~DTC();
    void setHorasInv();
    int getHorasInv();
    void setSalario();

protected:
private:
};

class DHC: public Docente
{
public:
    int cantHorasTrab;
    string nivelEstudios;

    DHC();
    virtual ~DHC();
    void setHorasTrab();
    void setNivelEstudios();
    int getHorasTrab();
    string getNivelEstudios();
    void setSalario();

protected:
private:
};
```

Ejemplo completo:  
DocentesHerencia.zip