



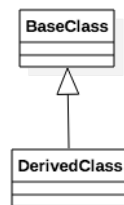
# Programação Orientada a Objetos

Prof. Hugo Marcondes  
hugo.marcondes@ifsc.edu.br

Aula 07

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina

- Herança permite a definição de uma classe em termos de outra classe
  - Classe Base
  - Classe Derivada



`class` DerivedClass: `access-specifier` BaseClass

# Exemplo



INSTITUTO FEDERAL  
SANTA CATARINA

```
// Base class
class Shape {
public:
    void setWidth(int w) {
        width = w;
    }
    void setHeight(int h) {
        height = h;
    }
protected:
    int width;
    int height;
};

// Derived class
class Rectangle: public Shape
{
public:
    int getArea() {
        return (width * height);
    }
};
```

```
int main(void)
{
    Rectangle Rect;

    Rect.setWidth(5);
    Rect.setHeight(7);

    // Print the area of the object.
    cout << "Total area: " << Rect.getArea() << endl;

    return 0;
}
```

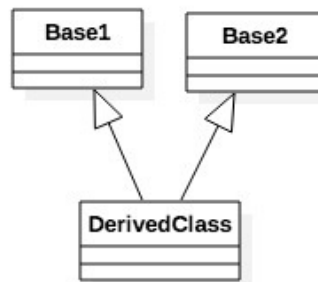


- Uma classe derivada herda todos os atributos e métodos da classe base, com as seguintes exceções
- Construtores, Destrutores e Construtores de cópia da classe Base
- Operadores sobrecarregados da classe Base
- Funções “amigas” (friend functions) da classe Base

- Os atributos e métodos de uma classe podem ser qualificadas de acordo com a sua acessibilidade.
- Os qualificadores de acesso em C++ podem ser sumarizados de acordo com a tabela abaixo:

Acesso	public	protected	private
Interno	✓	✓	✓
Classe Derivada	✓	✓	✗
Externo	✓	✗	✗

- Em C++ as heranças também podem ser qualificadas com qualificadores de acesso
  - Determinam a visibilidade dos métodos herdados pela classe Base, na classe Derivada
- Herança Pública (**public**)
  - Não alteram os qualificadores dos métodos da classe Base, na classe Derivada.
- Herança Protegida (**protected**)
  - Os métodos público e protegidos da classe Base, se tornam métodos protegidos na classe Derivada
- Herança Privada (**private**)
  - Os métodos públicos e protegidos da classe Base, se tornam métodos privados na classe Derivada



`class` **DerivedClass**: `access` **Base1**, `access` **Base2**, ...

# Exemplo



INSTITUTO FEDERAL  
SANTA CATARINA

```
// Base class Shape
class Shape {
public:
    void setWidth(int w) {
        width = w;
    }
    void setHeight(int h) {
        height = h;
    }
protected:
    int width;
    int height;
};

// Base class PaintCost
class PaintCost {
public:
    int getCost(int area) {
        return area * 70;
    }
};
```

```
// Derived class
class Rectangle: public Shape, public PaintCost {
public:
    int getArea() {
        return (width * height);
    }
};

int main(void) {
    Rectangle Rect;
    int area;

    Rect.setWidth(5);
    Rect.setHeight(7);

    area = Rect.getArea();
    // Print the area of the object.
    cout << "Total area: " << Rect.getArea() << endl;

    // Print the total cost of painting
    cout << "Total paint cost: $";
    cout << Rect.getCost(area) << endl;

    return 0;
}
```