

# Programación Orientada a Objetos

## Tarea 2.2. Constructores y uso de objetos

José Fidel Argudo Argudo    Francisco Palomo Lozano  
Inmaculada Medina Bulo    Gerardo Aburrizaga García



Versión 1.0



## Tarea 2.2. Cuestiones

### Ejercicio 1

¿Se puede definir un constructor con un nombre diferente al de la clase? Justifique la respuesta.

## Tarea 2.2. Cuestiones

### Ejercicio 2

Enumere las diferencias existentes entre inicializar un atributo en la lista de inicialización y asignarle un valor en el cuerpo del constructor. ¿Es posible utilizar dicha lista en otros métodos de una clase?

## Tarea 2.2. Cuestiones

### Ejercicio 3

```
1 class punto {
2     double x, y;
3 public:
4     punto(double a = 0., double b = 0.) : x{a}, y{b} {}
5     punto(const punto &p) : x{p.x}, y{p.y} {}
6     punto& operator =(const punto &p)
7     { x = p.x; y = p.y; return *this; }
8 };
```

Diga qué función de la clase punto se llama en cada una de las siguientes líneas, o si es incorrecta, suponiendo que se han ejecutado las anteriores, corregidas si es necesario.

1 punto p;	5 punto t{};
2 punto q();	6 punto u(q);
3 punto r(2.,);	7 punto v = r;
4 punto s{3.4};	8 t = s;

## Tarea 2.2. Cuestiones

### Ejercicio 4

Dadas las clases Libro1 y Libro2

```
1 class Libro1 {
2     string titulo_; int pags_;
3 public:
4     Libro1(string t="",
5           int p=0);
6     // ...
7 };
8 class Libro2 {
9     string titulo_; int pags_;
10 public:
11     Libro2(string t, int p=0);
12     Libro2(const char* c);
13     // ...
14 };
```

decida si x se puede sustituir por 1 o 2 en los siguientes items:

- 1 Se puede definir: LibroX lib1;
- 2 Se tiene un constructor de conversión de std::string a LibroX
- 3 Se puede definir: LibroX lib2[5];

## Tarea 2.2. Cuestiones

### Ejercicio 4 (cont.)

```
1 class Libro1 {
2     string titulo_; int pags_;
3 public:
4     Libro1(string t="",
5             int p=0);
6     // ...
7 };

8 class Libro2 {
9     string titulo_; int pags_;
10 public:
11     Libro2(string t, int p=0);
12     Libro2(const char* c);
13     // ...
14 };
```

- ④ Se puede definir: `std::vector<LibroX> lib3;`
- ⑤ Se produce una conversión implícita de `const char*` a `string` al ejecutar `LibroX* lib4 = new LibroX("El Quijote");`
- ⑥ Se puede definir: `LibroX lib5 = "El Quijote";`

## Tarea 2.2. Cuestiones

### Ejercicio 5

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;

5  class Libro {
6      char* titulo_; int paginas_;
7  public:
8      Libro() : titulo_(new char[1]), paginas_(0) { *titulo_ = 0; }
9      Libro(const char* t, int p) : paginas_(p) {
10         titulo_ = new char[strlen(t) + 1];
11         strcpy(titulo_, t);
12     }
13     ~Libro() { delete[] titulo_; }
14     void paginas(int p) { paginas_ = p; }
15     int paginas() const { return paginas_; }
16     char* titulo() const { return titulo_; }
17 };
```

## Tarea 2.2. Cuestiones

### Ejercicio 5 (cont.)

```
18 void mostrar(Libro l) {
19     cout << l.titulo() << " tiene "
20         << l.paginas() << " páginas" << endl;
21 }
22 int main() {
23     Libro l1("Fundamentos de C++", 474),
24           l2("Por Fin: C ISO", 224),
25           l3;

27     l3 = l1;
28     mostrar(l1), mostrar(l2), mostrar(l3);
29 }
```

Diga si el programa funciona correctamente. En caso afirmativo indique lo que imprime. En caso negativo haga las modificaciones necesarias para que funcione correctamente.



## Tarea 2.2. Cuestiones

### Ejercicio 6

¿Se consigue llamar a la función `f` o se producirá un error? En tal caso, corríjalo.

```
1 struct B;
2 struct A {
3     A(B);
4     // ...
5 };
6 struct B {
7     operator A();
8     // ...
9 };

11 void f(A&);

13 main() { B b; f(b); }
```