

# Programación Orientada a Objetos

## Tarea 2.1. Estudio de la estructura de clase

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



Versión 2.1



## Tarea 2.1. Cuestiones

### Ejercicio 1

¿Cómo implementa el lenguaje de programación C++ el principio de encapsulamiento?

### Ejercicio 2

¿Cómo implementa el lenguaje de programación C++ el principio de ocultación de información?

## Tarea 2.1. Cuestiones

### Ejercicio 3

¿Hay algún error en el siguiente programa? Si es así, explique por qué y corríjalo.

```
1  #include <iostream>
2  class C {
3  public:
4      C(int i = 0, bool al = false): n(i), hayAlerta(al) {}
5      void mostrar() {
6          if(hayAlerta) alerta();
7          std::cout << "i_=" << n << std::endl;
8      }
9  private:
10     int n;
11     bool hayAlerta;
12     void alerta() { std::cout << "Alerta_" << std::endl; }
13 };

15 int main() { const C c; c.mostrar(); }
```

## Tarea 2.1. Cuestiones

### Ejercicio 4

Indique los errores que hay en el siguiente código y su causa.

```
1  class C {  
2  public:  
3      C();  
4      C(int a, int b, int c, int d);  
5      int f1(int i) const;  
6      int f2(int i);  
7      static void f3() {m = h;}  
8      static int n;  
9  private:  
10     int h;  
11     mutable int i;  
12     const int j;  
13     mutable int k;  
14     static int m;  
15 };
```

## Tarea 2.1. Cuestiones

### Ejercicio 4 (cont.)

```
17 int C::f1(int i) const
18 {
19     h = i; k = i; return 0;
20 }

22 C::C() {h = i = j = k = 0;}

24 C::C(int a, int b, int c, int d) : h(a), i(b), j(c), k(d) {}

26 int C::f2(int i)
27 {
28     k = i; h = i; return 0;
29 }
```

## Tarea 2.1. Cuestiones

### Ejercicio 5

Considerando el código de la pregunta anterior, determine si hay errores en el código de esta pregunta. ¿Falta algo para que podamos llegar a ejecutar este programa? En tal caso, haga las correcciones necesarias para solucionarlo e indique qué imprime el programa.

```
1  #include <iostream>
2  class C { // clase C anterior
3      //...
4  };
5  // definición de métodos de la clase C aquí...

7  int main()
8  {
9      C c1, c2;
10     C::n = 3;
11     c1.n = 4;
12     std::cout << C::n << "□" << c2.n << std::endl;
13 }
```

## Tarea 2.1. Cuestiones

### Ejercicio 6

Considere la siguiente clase:

```
1 class C {  
2     public:  
3         C(int i) : n(i), m(0.0) {} // ctor. de conversión  
4         // ...  
5     private:  
6         int n;  
7         double m;  
8 };
```

Escriba las declaraciones correspondientes a la sobrecarga del operador + para objetos de tipo C como miembro y como función externa. A continuación, escriba a modo de ejemplo un trozo de código que provoque un error de compilación con el operador miembro, pero no con el externo.