

1. Cree una clase A con un constructor por defecto inline que se anuncia a sí mismo. Ahora cree una nueva clase B y ponga un objeto de A como miembro de B y dé a B un constructor inline que también se anuncie. Cree un arreglo de objetos B y vea qué ocurre.
2. Agregue en el ejercicio anterior destructores que también se anuncien. ¿Qué puede observar con respecto al orden de construcción y destrucción de objetos?
3. ¿Cuán grande es una estructura? Escriba un programa que imprima el tamaño de una estructura. Defina una estructura que tenga solo datos miembros y otra similar que además contenga funciones miembro, por último defina una estructura que no tenga datos y solo tenga funciones miembro. Imprima el tamaño de cada estructura. Justifique los resultados.
4. Cree una clase sin constructor y muestre que se pueden crear objetos utilizando el constructor por defecto. Ahora cree un constructor que reciba un argumento y reintente el mismo **main**.
5. Cree una clase **Message** con un constructor que tome un argumento del tipo **string** con un valor por defecto. Cree un miembro privado del tipo **string** e inicialice el mismo con el argumento recibido en el constructor. Cree dos funciones sobrecargadas llamadas **print**, una que no reciba argumentos e imprima el miembro **string** y otro que reciba un argumento del tipo **string** que se imprima juntamente con el miembro. Tiene sentido realizar esto o sería mejor utilizar un argumento por defecto como en el constructor?
6. Cree una nueva versión de la clase **Stack** que contenga un constructor por defecto y un segundo constructor que reciba un vector de punteros a objetos y el tamaño del vector. Este constructor debería recorrer el arreglo e ir **pusheando** cada puntero sobre el stack. Probar la clase con un arreglo de punteros a **string**.
7. Cree una clase con un destructor que imprima un mensaje y luego llame a **exit()**. Cree un objeto global de esta clase y otro en **main** y vea qué ocurre.