

1. Cree una clase A con un constructor por defecto que se anuncie 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 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. 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.
4. Comprobar la existencia de constructores de copia implícitos.
5. Comprobar la activación constructores de copia cuando:
 - a. Se crean objetos como copia de otros del mismo tipo.
 - b. Se pasan argumentos por valor a una función.
 - c. Se retorna un objeto de un UDT.
6. Comprobar que cuando una función recibe objetos por referencia, el “copy constructor” no es activado.
7. Hacer un programa que agregue a un `vector<T> v;` 10 elementos de tipo `T`. Redimensione el vector a través del método `vector<T>::resize` para llevarlo a 3 elementos. Verifique que los destructores de los 7 elementos eliminados son activados. Que pasa si ahora lo redimensiona nuevamente a 10 elementos?
8. Cree una clase con un constructor de copia que se anuncie a sí mismo (en `cout`). Haga una segunda clase conteniendo un objeto miembro de la clase anterior pero no cree un constructor de copia. Muestre que el constructor sintetizado en la segunda clase llama automáticamente al constructor de copia de la primera clase.