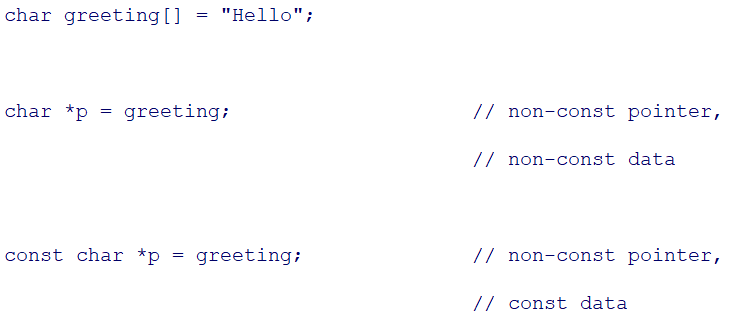
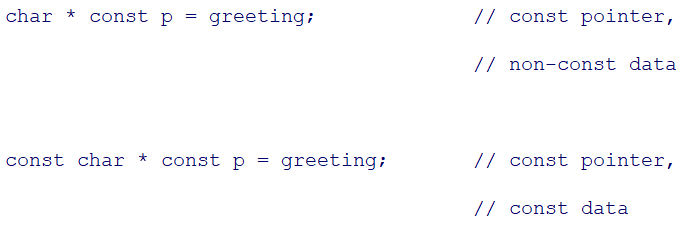
Item 3 Usar const siempre que sea posible

Lo maravilloso acerca de const es que le permite a usted especificar una restricción semántica a un objeto particular que no debe ser modificado y los compiladores forzarán esa restricción. El uso de const le permite a usted notificar a compiladores y programadores que un valor debe permanecer invariante. Siempre que eso sea cierto, usted debe asegurarse de decir que así es, porque de esa forma usted enlista la ayuda de sus compiladores para asegurar que la restricción no es violada.

La palabra reservada const es muy versátil. Fuera de las clases, usted puede usarla para constantes en alcance global o de espacio de nombres (Item 2 de [Meyers]), así como objetos declarados static en alcances de archivo, de función, o de bloque. Dentro de las clases, usted puede usarla con ambos, atributos estáticos y no estáticos. Para apuntadores, usted puede especificar si el apuntador mismo es constante, el dato al que apunta es constante, ambos son constantes, o ninguno es constante:





Esta sintaxis no es tan caprichosa como podría parecer. Si la palabra const aparece a la izquierda del asterisco, lo que es apuntado es constante; si la palabra const aparce a la derecha del asterisco, el apuntador mismo es constante; si const aparece a ambos lados, ambas cosas son constantes.

Cuando lo que es apuntado es constante, algunos programadores ponen const antes del tipo de dato. Otros lo ponen después del tipo de dato, pero antes del asterisco. No hay diferencia en el significado, así que las siguientes funciones toman el mismo tipo de parámetro:





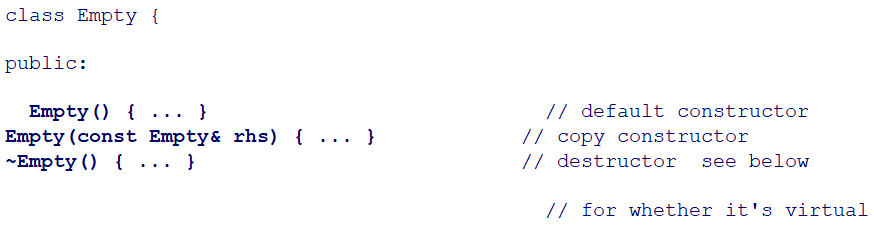
Dado que ambas formas existen en código real, usted debe acostumbrarse a ambas.

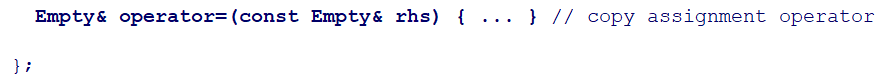
Item 5: Saber qué funciones C++ silenciosamente escribe y llama ([Meyers, 116,0/714])

¿Cuándo una clase vacía no es una clase vacía? Cuando C++ se encarga de ella. Si usted mismo no los declara, los compiladores declararán sus propias versiones de un constructor de copia, un operador de asignación de copia, y un destructor. Además, si usted no declara ningún constructor, los compiladores también declararán un constructor por defecto por usted. Todas estas funciones serán públicas e inline. Como resultado, si usted escribe

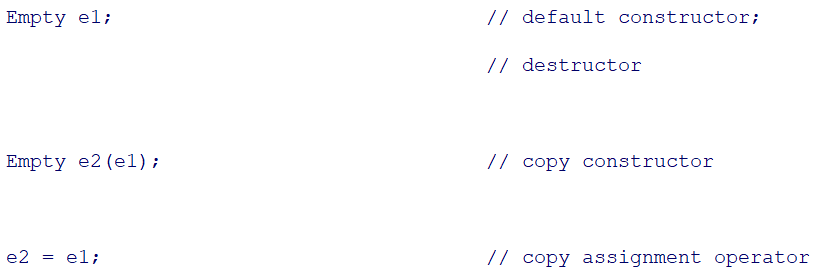


esto es esencialmente lo mismo que si usted hubiese escrito esto:





Esas funciones son generadas solamente si se necesitan, pero no se requiere hacer mucho para que sean necesarias. El siguiente código hará que cada una de las funciones sean generadas:



REF.

[Meyers] Effective C++, 3rd Edition, 55 Specific Ways to Improve Your Programs and Designs.