

¿En. C++, porqué debemos usar 'delete' con objetos creados mediante 'new'?

Seleccione una:

- O Porque si no lo hacemos provocaremos tarde o temprano un error de 'violación de segmento' ('segmentation fault')
 - No es necesario, solo es conveniente hacerlo.
 - O
 - O Porque nos obliga C++.
 - Porque es responsabilidad del programador administrar la memoria dinámica del programa.

```
Fecha f3(f1);
} else if ( ... ) {
   Fecha *f4 = new Fecha(f1);
  f5=f4;
 delete f5; f5=NULL;
// ¿Cuántos objetos 'Fecha' hay en este punto del código?
Respuesta: 1
```

Fecha f1(12,9,2016);

} else if (...) {

Fecha *f5; if (...) { Fecha f2(f1);

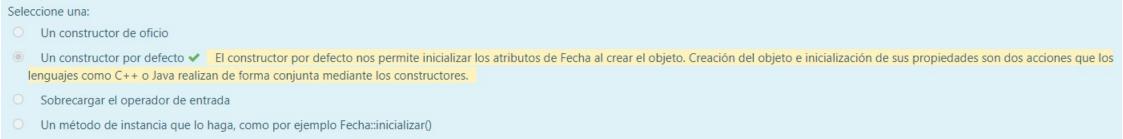
. . .

Sólo f1 ha sobrevivido, f2 y f3 han sido destruídos al terminar el bloque donde se habían definido, f4 y f5 son dos punteros apuntando al mismo objeto, que es destruído al hacer 'delete'.

¿Dónde se guarda el valor de un atributo de instancia de un objeto que está en memoria? Seleccione una: En la memoria estática. En el disco duro. En la memoria que se reservó para el objeto al crearlo. En una zona especial de la memoria destinada al almacenamiento de atributos, distinta de donde se guardan los objetos y sus punteros

En la memoria que se reservó para la clase del objeto al crearlo.

¿Por qué ocultamos los datos (atributos) en una clase?		
Color	rainna una a más da una	
Seleccione una o más de una:		
	Para poder usar los nombres de atributos que queramos sin tener que publicarlos.	
V	Para evitar que el código cliente pueda manipular el estado de nuestros objetos de forma que queden en un estado inconsistente o corrupto (p. ej., que el día de una fecha sea negativo o mayor que 31).	
	Para poder tener atributos que sólo usamos internamente en los métodos (funciones) de la clase pero que el código cliente no necesita.	
V	Para ocultar los detalles de implementación de nuestra clase al código cliente. ✓	



¿Cómo podemos crear objetos Fecha con valores iniciales diferentes?		
2 8		
Seleccione una:		
	Implementando un método de instancia, p. ej. Fecha::set(int dia, int mes, int anyo).	
	Sobrecargando el operador de entrada.	
	Implementando un constructor de copia.	
•	Implementando un constructor sobrecargado. 🗸 Aunque Fecha:set(int dia, int mes, int anyo) hace practicamente el mismo trabajo que el constructor sobrecargado (de hecho, podríamos llamar al	
	método desde el constructor), éste lo hace justo en el momento en el que se crea el objeto, garantizando que el objeto 'nace' con esos valores iniciales, antes de que podamos hacer uso de él de alguna	
	forma, cosa que no está garantizada con el método set().	

¿Cómo podemos crear fechas que sean copia de otras? Seleccione una: Implementando un constructor sobrecargado. Implementando un constructor de copia. Implementando un método que haga la copia, como Fecha::copiar(Fecha f)

Implementando un constructor sobrecargado que luego usamos para copiar la fecha: Fecha f2(f1.getDia(), f1.getMes(), f1.getAnyo()).

Cuando decimos que un objeto que hemos creado 'ha sido eliminado', 'ya no existe', ¿qué estamos diciendo exactamente?

Seleccione una:

Que el objeto ha sido eliminado de la memoria de nuestro programa y ya no es accesible.
 Que la variable mediante el cual nos referimos a él va no existe porque hemos salido del ámbito donde fue definida.

Que hemos hecho 'delete'.

¿Qué pasa cuando un objeto 'desaparece'?

Se ejecuta el destructor de la clase sobre ese objeto (C++) 🗸

Seleccione una o más de una:

■ Es eliminado de la memoria de nuestro programa. ✓

Todas las variables que hacen referencia a él (punteros, referencias) ya no se pueden usar.