

Dynamische Speicherplatzverwaltung mit new / delete

When **new** is used to **allocate memory** for a C++ class object, the object's **constructor** is called after the memory is allocated.

Use the **delete** operator to **deallocate the memory** allocated with the **new** operator.

When allocating a **multidimensional array**, all dimensions **except the first** must be **constant expressions** that evaluate to positive values; the leftmost array dimension can be any expression that evaluates to a positive value.

When allocating an array using the **new** operator, the first dimension can be **zero** — the **new** operator returns a **unique pointer**.

```
char (*pchar)[10] = new char[dim][10]; delete [] pchar;
```

The *type-name* cannot contain **const**, **volatile**, **class declarations**, or **enumeration declarations**. Therefore, the following expression is illegal:

```
volatile char *vch = new volatile char[20];
```

The **new** operator does not allocate **reference types** because they are **not objects**.

Dynamische Speicherplatzverwaltung mit new / delete

The **new** operator **cannot** be used to **allocate a function**, but it can be **used to allocate pointers to functions**.

The following example allocates and then frees an **array of seven pointers to functions that return integers**.

```
int(**p) () = new(int(*[7]) ()); delete *p;
```