|  |
| --- |
| Circle Language Spec: System Objects |

## Cross-Aspect Assignments

The standard way to use the Class aspect in an assignment is to get the Object aspect from one reference and assign it to the Class aspect of another reference. But you can also do it the other way around: get the Class aspect from one reference and assign it as the Object aspect of another reference. You can call it *Class-to-Object* assignment. It can also be called a *Class-Get* *assignment*. Less conventional ways of yielding over aspects like that, is also called a *cross-aspect* assignment.

Also note here, that there are two ways to get the Class aspect: Get the Class aspect of the *object* or Get the Class aspect of the *reference*.

|  |
| --- |
| Reference-Class to Object Assignment |
|  |
| Reference-Class Get 🡨  Object Set 🡪 |
| *Result:* |
| *The object reference on the right now points to the class of the object reference on the left.* |
|  |
|  |
| Object-Class to Object Assignment |
|  |
| Object-Class Get 🡨  Object Set 🡪 |
| *Result:* |
| *The object reference on the right now points to the class of the object on the left.* |

If the source of the assignment is a pointer-to-pointer, then the target also becomes a pointer-to-pointer. So this also gives Reference Class to Object assignment the following implementations:

|  |
| --- |
| Reference-Class to Object Assignment |
|  |
| Reference-Class Get 🡨 (~= Other Related Item Class Get)  Object Set 🡪 (~= Other Related Item Set) |
| *Result:* |
|  |
|  |
|  |
| Reference-Class to Object Assignment |
|  |
| Reference-Class Get 🡨 (~= Other Related List Item Class Get)  Object Set 🡪 (~= Other Related List Item Set) |
| *Result:* |
|  |

### Cross-Aspect Pointer Assignments

*Pointer* assignments do not have a cross-aspect variation. Pointer assignments use an the reference aspect as the source of an assignment: not a particular aspect of the object reference, but the reference itself. It does not apply to cross-aspect assignments, because on one end of the assignment no aspect at all is involved.