The new-expression

Syntax of the new expression

The new-expression has the form:

new constructor-call

Here is an example:

```
new C(5)
```

The three steps in evaluating a new-expression

We present the algorithm for evaluating a new-expression. In describing its evaluation, we don’t have access to the class or constructor, so our algorithm is in terms of the general form of the new-expression.

Here we go:

1. Create (or draw) an object of the class named by the constructor-call, with default values for the fields.

   (As you know, the default value for a field depends on its type. For example the default for type **int** is 0, and the default for type **boolean** is **false**.)

2. Execute the constructor call.

   (The goal of the constructor call is to truthify the class invariant, but at this abstract level we don’t know what it will do.)

3. Give as value of the new-expression the name of (that is, the pointer to) the object.

   (Remember, that name was placed on the object when it was created.)

Example of evaluation of a new-expression

We give an example. Here is a definition of class C, with two fields and a constructor. We evaluate the new-expression C(5).

1. Create an object of class C with default values for the fields.

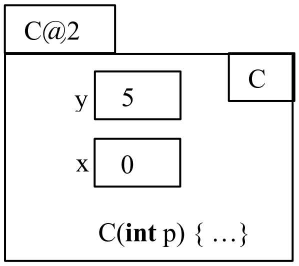
   Since *we* are creating the object, *we* choose the name.
   We choose C@2.

2. Execute the constructor call C(5), which stores 5 in y.

3. Give as value of the new-expression the name of the object, C@2.

So, this evaluation of C(5) resulted in the value C@2.

```
public class C {
   int x;
   int y;
   public C(int p) {
      y= p;
   }
}
```



This shows the object after evaluation of the new-expression is done