**class exit — Exit class-member program and return result**

**Description**

`class exit` exits a class-member program and optionally returns the specified result.

`class exit` may be used only from class-member programs; see [P] class.

**Syntax**

```
class exit [ rvalue ]
```

where `rvalue` is

```
" [ string ] "
" " [ string ] ",
#
exp
(exp)
.id[ .id[ . . . ] ] [ program_arguments ]
{}
{el[ ,el[ , . . . ] ]}
```

See [P] class for more information on `rvalues`.

**Remarks and examples**

Do not confuse returned values with return codes, which all Stata programs set, including member programs. Member programs exit when they execute.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Returned value</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>class exit with arguments</code></td>
<td>as specified</td>
<td>0</td>
</tr>
<tr>
<td><code>class exit without arguments</code></td>
<td>nothing</td>
<td>0</td>
</tr>
<tr>
<td><code>exit without arguments</code></td>
<td>nothing</td>
<td>0</td>
</tr>
<tr>
<td><code>exit with arguments</code></td>
<td>nothing</td>
<td>as specified</td>
</tr>
<tr>
<td><code>error</code></td>
<td>nothing</td>
<td>as specified</td>
</tr>
<tr>
<td><code>command having error</code></td>
<td>nothing</td>
<td>as appropriate</td>
</tr>
</tbody>
</table>

Any of the preceding are valid ways of exiting a member program, although the last is perhaps best avoided. `class exit without arguments` has the same effect as `exit without arguments`; it does not matter which you use.
Examples

```plaintext
class exit sqrt((.c0.y1-.c1.y0)^2 + (.c0.y1-.c1.y0)^2)
class exit "myresult"
class exit (.)
class exit "true"
class exit { ‘one’, ‘two’ }
class exit .coord
class exit .coord.x

tempname a
...
class exit .‘a’
```

Warning: Distinguish carefully between “class exit .a” and “class exit (.a)”. The first returns a copy of the instance .a. The second returns a double equal to the extended missing value .a.

Also see

[P] class — Class programming
[P] exit — Exit from a program or do-file
[M-2] class — Object-oriented programming (classes)