TypeInfo

COLLABORATORS			
	TITLE : TypeInfo		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 6, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

TypeInfo

1.1 Implementation notes

```
                    The TypeInfo class
                ----------------------------
                ($Date: 1994/08/01 15:57:36 $)

The Type_info class introduces a runtime type inquiry mechanism as described
by Bjarne Stroustrup in his book "The C++ Programming Language".

IMPORTANT NOTE: With GCC 2.5.8. the Type_info system works only for
single inheritance lines! Multiply inherited virtual Type_info methods
are not treated correctly!!
Second, you MUST define Type_info methods and static info_obj for each
of your derived classes that ought to work with it. When the virtual
methods are not being overridden, GCC produces code that again does not
call the method that overrode last!


How to provide Type_info support to your class

------------------------------------------------------------
 -> Back to the root menu..
```

1.2 typeinfo_1

```
Every class that wants to get the Type_info support has to define a static
data member, the 'info_obj' where the class information is laid down in
addition to the Type_info inquiry methods.

Add the following to your class header file:

public:
   // each class within the 'Type_info' system needs to declare..
   static const Type_info info_obj;     // the per-class data
   virtual const Type_info& get_info() const
      { return info_obj; }
   static const Type_info& info()
```

```
      { return info_obj; }
```

and insert the static object definition in your class source file:

```
typeinfo(MyClass, derived(from(class_A) from(class_B)) ,"$VER: Version 68.23 ↩
   (23.02.94)$");
```

This macro, defined in 'APlusPlus/environment/TypeInfo.h', initialises
the static Type_info info_obj. The runtime type inquiry mechanism works
only on base of the information provided with this macro, especially
the list of direct base classes.

IMPORTANT NOTE: With GCC 2.5.8. the Type_info system works only for
single inheritance lines! Multiply inherited virtual Type_info methods
are not treated correctly!!
Second, you MUST define Type_info methods and static info_obj for each
of your derived classes that ought to work with it. When the virtual
methods are not being overridden, GCC produces code that again does not
call the method that overrode last!