Resolution of Mantis items 2575, 2598, 2608: specifying access to class parameters and type parameters.

Ballot ids: 50, 52, 55, 59, 64

Section **8.5**

Change title of 8.5:

REPLACE:
8.5 Object properties

WITH:
8.5 Object properties and object parameter data

After : There are no restrictions on the data type of a class property.

Add:

The parameter data values of an object can also be accessed by qualifying the class parameter or local parameter names with an instance name. Example:

```
class vector #(parameter width = 7);
endclass

vector #(3)  v = new;
initial $display (v.width);
```

Such an expression is not a constant expression.

Section **8.10 This**

REPLACE:

The **this** keyword is used to unambiguously refer to class properties or methods of the current instance.

WITH:
The **this** keyword is used to unambiguously refer to class properties, parameters, local params or methods of the current instance.
.

Section **8.14 Super**

REPLACE:

The **super** keyword is used from within a derived class to refer to members of the base class. It is necessary to use **super** to access members of a base class when those members are overridden by the derived class.

WITH:
The **super** keyword is used from within a derived class to refer to members, class parameters or local parameters of the base class. It is necessary to use **super** to access members or parameters of a base class when those ~~members~~ are overridden by the derived class. A parameter expression using super to access the parameter is not a constant expression.
.
REPLACE:
The member can be a member declared a level up or be inherited by the class one level up.

WITH:
The member or parameter can be ~~a member~~ declared a level up or be inherited by the class one level up

Section **8.17**

REPLACE:

In SystemVerilog, unqualified class properties and methods are public, available to anyone who has accessto the object's name.

WITH:
In SystemVerilog, unqualified class properties and methods are public, available to anyone who has accessto the object's name. Class parameters and class local parameters are also public.

Section **8.22**

In 8.22, REPLACE
Because classes and other scopes can have the same identifiers, the class scope resolution operator uniquely identifies a member of a particular class. In addition to disambiguating class scope identifiers, the :: operator also allows access to static members (class properties and methods) from outside the class, as well as access to public or protected elements of a superclass from within the derived classes.

WITH

Because classes and other scopes can have the same identifiers, the class scope resolution operator uniquely identifies a member of a particular class, a class parameter, class type parameter or class local parameter. In addition to disambiguating class scope identifiers, the :: operator also allows access to static members (class properties and methods , class parameters, class type parameters and class local parameters from outside the class, as well as access to public or protected elements of a superclass from within the derived classes. A class parameter, type parameter or local param is a public element of a class. A class scope parameter or class scope type parameter is a constant expression.

REPLACE:
In SystemVerilog, the class scope resolution operator applies to all static elements of a class: static class properties, static methods, typedefs, enumerations, structures, unions, and nested class declarations.

WITH
In SystemVerilog, the class scope resolution operator applies to all static elements of a class: static class properties, static methods, typedefs, enumerations, parameters, type parameters, local parameters, constraints, covergroups, structures, unions, and nested class declarations.

REPLACE:
The class scope resolution operator enables the following:
— Access to static public members (methods and class properties) from outside the class hierarchy.
— Access to public or protected class members of a superclass from within the derived classes.
— Access to type declarations and enumeration named constants declared inside the class from outside the class hierarchy or from within derived classes.

WITH:
The class scope resolution operator enables the following:
— Access to static public members (methods and class properties) from outside the class hierarchy.
— Access to public or protected class members of a superclass from within the derived classes.
— Access to constraint, covergroup, type declarations and enumeration named constants declared inside the class from outside the class hierarchy or from within derived classes.
— Access to parameters, type parameters and local params declared inside the class from outside the class hierarchy or from within derived classes.

REPLACE:
Nested classes shall have the same access rights as methods do in the containing class. They have full access rights to **local** and **protected** methods and properties of the containing class. Nested classes have lexically-scoped, unqualified access to the **static** properties and methods of the containing class

WITH:
Nested classes shall have the same access rights as methods do in the containing class. They have full access rights to **local** and **protected** methods and properties of the containing class. Nested classes have lexically-scoped, unqualified access to the **static** properties and methods, parameters, type parameters and local parameters of the containing class.

In BOTH Syntax 11-8 in section 11.12:

AND

Section **A.8.4 Primaries**

REPLACE
constant_primary ::=
primary_literal
| ps_parameter_identifier constant_select
| specparam_identifier [**[** constant_range_expression **]**]
| genvar_identifier35
| [package_scope | class_scope] enum_identifier
| constant_concatenation [**[** constant_range_expression **]**]
| constant_multiple_concatenation [**[** constant_range_expression **]**]
| constant_function_call
| constant_let_expression

| (constant_mintypmax_expression)
| constant_cast
| constant_assignment_pattern_expression

| type_reference3

WITH:
constant_primary ::=
primary_literal
~~| ps_parameter_identifier constant_select~~
| [package_scope | class_scope] parameter_identifier constant_select
| specparam_identifier [[constant_range_expression]]
| genvar_identifier35
| [package_scope | class_scope] enum_identifier
| constant_concatenation [[constant_range_expression]]
| constant_multiple_concatenation [[constant_range_expression]]
| constant_function_call
| constant_let_expression
| (constant_mintypmax_expression)
| constant_cast
| constant_assignment_pattern_expression

| type_reference3
Because classes and other scopes can have the same identifiers, the class scope resolution operator uniquely identifies a member of a particular class. In addition to disambiguating class scope identifiers, the :: operator also allows access to static members (class properties and methods) from outside the class, as well as

access to public or