OOP Changelog

Changes to the PHP OOP model are logged here. Descriptions and other notes regarding these features are documented within the OOP model documentation.

ENTITY	ENTITY
8.5.0	Changed: A method that returns `static` may now be overridden in a final class to declare `self` or the class's own type as its return type, as these are equivalent to `static` in a class that cannot be extended further.
8.5.0	Changed: Traits are now bound to a class before its parent class.
8.4.0	Added: Support for Property Hooks.
8.4.0	Added: Support for Lazy Objects.
8.1.0	Added: Support for the final modifier for class constants. Also, interface constants become overridable by default.
8.0.0	Added: Support for the nullsafe operator *?-ENTITY* to access properties and methods on objects that may be null.
7.4.0	Changed: It is now possible to throw an exception within `__toString`.
7.4.0	Added: Support for limited return type covariance and argument type contravariance. Full variance support is only available if autoloading is used. Inside a single file only non-cyclic type references are possible.
7.4.0	Added: It is now possible to type class properties.
7.3.0	Incompatibility: Argument unpacking of `Traversable`s with non-ENTITY keys is no longer supported. This behaviour was not intended and thus has been removed.
7.3.0	Incompatibility: In previous versions it was possible to separate the static properties by assigning a reference. This has been removed.
7.3.0	Changed: The instanceof operator now allows literals as the first operand, in which case the result is always ENTITY.
7.2.0	Deprecated: The `__autoload` method has been deprecated in favour of `spl_autoload_register`.
7.2.0	Changed: The following name cannot be used to name classes, interfaces, or traits: `object`.

ENTITY	ENTITY
7.2.0	Changed: A trailing comma can now be added to the group-use syntax for namespaces.
7.2.0	Changed: Parameter type widening. Parameter types from overridden methods and from interface implementations may now be omitted.
7.2.0	Changed: Abstract methods can now be overridden when an abstract class extends another abstract class.
7.1.0	Changed: The following names cannot be used to name classes, interfaces, or traits: `void` and `iterable`.
7.1.0	Added: It is now possible to specify the visibility of class constants.
7.0.0	Deprecated: Static calls to methods that are not declared static.
7.0.0	Deprecated: PHP 4 style constructor. I.e. methods that have the same name as the class they are defined in.
7.0.0	Added: Group *use* declaration: classes, functions and constants being imported from the same namespace can now be grouped together in a single use statement.
7.0.0	Added: Support for anonymous classes has been added via `new class`.
7.0.0	Incompatibility: Iterating over a non-`Traversable` ENTITY will now have the same behaviour as iterating over by-reference ENTITYs.
7.0.0	Changed: Defining (compatible) properties in two used traits no longer triggers an error.
5.6.0	Added: The __debugInfo() method.
5.5.0	Added: The ::class magic constant.
5.5.0	Added: finally to handle exceptions.
5.4.0	Added: traits.
5.4.0	Changed: If an abstract class defines a signature for the constructor it will now be enforced.
5.3.3	Changed: Methods with the same name as the last element of a namespaced class name will no longer be treated as constructor. This change doesn't affect non-namespaced classes.
5.3.0	Changed: Classes that implement interfaces with methods that have default values in the prototype are no longer required to match the interface's default value.

ENTITY	ENTITY
5.3.0	Changed: It's now possible to reference the class using a variable (e.g., `echo $classname::constant;`). The variable's value can not be a keyword (e.g., `self`, `parent` or `static`).
5.3.0	Changed: An `E_WARNING` level error is issued if the magic overloading methods are declared static. It also enforces the public visibility requirement.
5.3.0	Changed: Prior to 5.3.0, exceptions thrown in the `__autoload` function could not be caught in the catch block, and would result in a fatal error. Exceptions now thrown in the __autoload function can be caught in the catch block, with one provision. If throwing a custom exception, then the custom exception class must be available. The __autoload function may be used recursively to autoload the custom exception class.
5.3.0	Added: The __callStatic method.
5.3.0	Added: heredoc and nowdoc support for class *const* and property definitions. Note: heredoc values must follow the same rules as double-quoted strings, (e.g. no variables within).
5.3.0	Added: Late Static Bindings.
5.3.0	Added: The __invoke() method.
5.2.0	Changed: The __toString() method was only called when it was directly combined with `echo` or `print`. But now, it is called in any ENTITY context (e.g. in `printf` with `%s` modifier) but not in other type contexts (e.g. with `%d` modifier). As of PHP 5.2.0, converting ENTITYs without a __toString method to ENTITY emits a `E_RECOVERABLE_ERROR` level error.
5.1.3	Changed: In previous versions of PHP 5, the use of `var` was considered deprecated and would issue an `E_STRICT` level error. It's no longer deprecated, therefore does not emit the error.
5.1.0	Changed: The __set_state() static method is now called for classes exported by `var_export`.
5.1.0	Added: The __isset() and __unset() methods.

□□□ #2
□ □□□ □□□ 2026-10-31 17:40:01 UTC
□ □□□ □□□ 2026-10-31 17:40:02 UTC