The body of class may optionally contain a single constructor declaration. Constructor declarations are described in section 8.3.

Member declarations are used to declare instance and static members of the class. Property member declarations are described in section 8.4 and index member declarations are described in section 8.5.

8.2 Members

The members of a class consist of the members introduced through member declarations in the class body and the members inherited from the base class.

8.2.1 Instance and Static Members

Members are either ***instance members*** or ***static members***.

Instance members are members of the class type (section 8.2.4) and its associated instance type. Within constructors, instance member functions, and instance member accessors, the type of `this` is the instance type (section 3.5.1) of the class.

Static members are declared using the `static` modifier and are members of the constructor function type (section 8.2.5). Within static member functions and static member accessors, the type of `this` is the constructor function type.

Class type parameters cannot be referenced in static member declarations.

8.2.2 Accessibility

Property members have either ***public*** or ***private*** accessibility. The default is public accessibility, but property member declarations may include a `public` or `private` modifier to explicitly specify the desired accessibility.

Public property members can be accessed everywhere, but private property members can be accessed only within the class body that contains their declaration. Any attempt to access a private property member outside the class body that contains its declaration results in a compile-time error.

Private accessibility is enforced only at compile-time and serves as no more than an *indication of intent*. Since JavaScript provides no mechanism to create private properties on an object, it is not possible to enforce the private modifier in dynamic code at run-time. For example, private accessibility can be defeated by changing an object’s static type to Any and accessing the member dynamically.

8.2.3 Inheritance and Overriding

A derived class ***inherits*** all members from its base class it doesn’t ***override***. Inheritance means that a derived class implicitly contains all non-overridden members of the base class. Both public and private property members are inherited, but only public property members can be overridden.