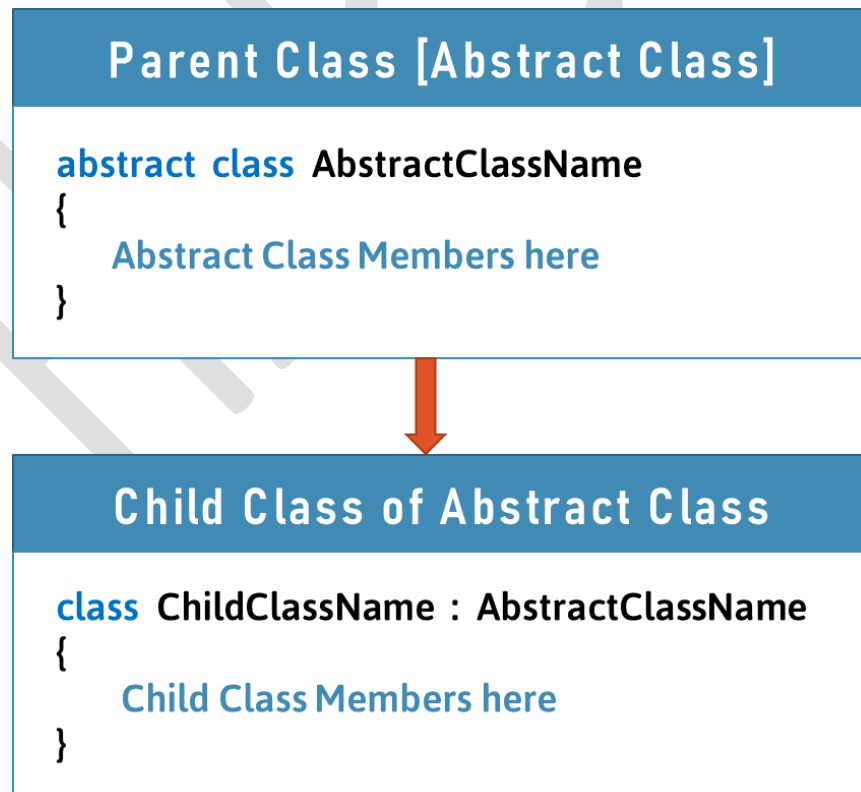
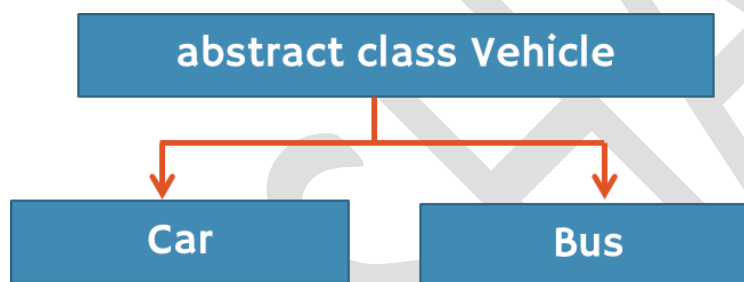


17

Abstract Classes and Abstract Methods

Introducing Abstract Classes

- › Abstract class is a parent class, for which, we can't create object; but we can create child classes.



- › The main intention of abstract class is to provide common set of fields and methods to all of its child classes of a specific group.
- › Abstract class can contain all types of members (fields, properties, methods, constructors etc.).
- › We can't create object for abstract class; but we can access its members through child class's object.
 - › So 'creating child class of abstract class' is the only-way to utilize abstract classes.
- › Use Abstract class concept, for the classes, for which, you feel creating object is not meaningful.

Class Type	Can Inherit from Other Classes	Can Inherit from Other Interfaces	Can be Inherited	Can be Instantiated
Normal Class	Yes	Yes	Yes	Yes
Abstract Class	Yes	Yes	Yes	No

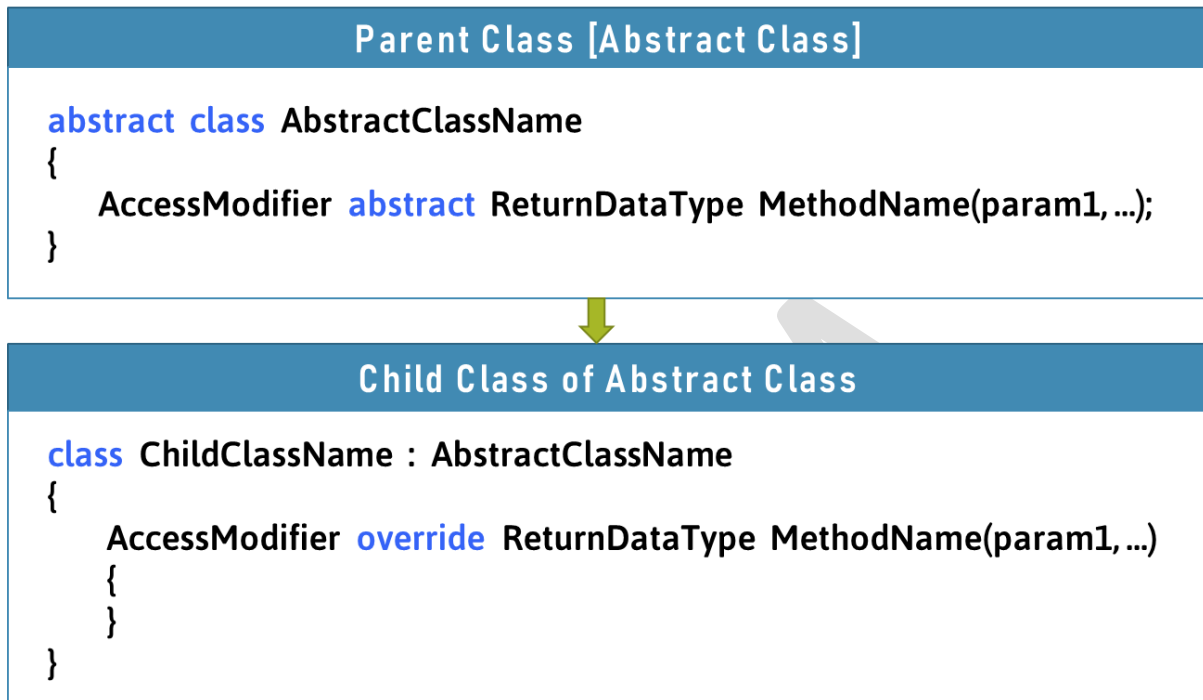
Type	1. Non-Static Fields	2. Non-Static Methods	3. Non-Static Constructors	4. Non-Static Properties	5. Non-Static Events	6. Non-Static Destructors	7. Constants
Normal Class	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Abstract Class	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Type	8. Static Fields	9. Static Methods	10. Static Constructors	11. Static Properties	12. Static Events	13. Virtual Methods	14. Abstract Methods	15. Non-Static Auto-Impl Properties	16. Non-Static Indexers
Normal Class	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Abstract Class	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Introducing Abstract Methods

- › Abstract methods are declared in parent class, with "abstract" keyword; implemented in child classes, with "override" keyword.
- › When the parent class don't want to provide the definition of a method;

- › it wants to let child classes to implement the method.



- › Abstract Methods contain "method declaration" only; but not "method body".
- › Child class must provide method body for abstract methods.

