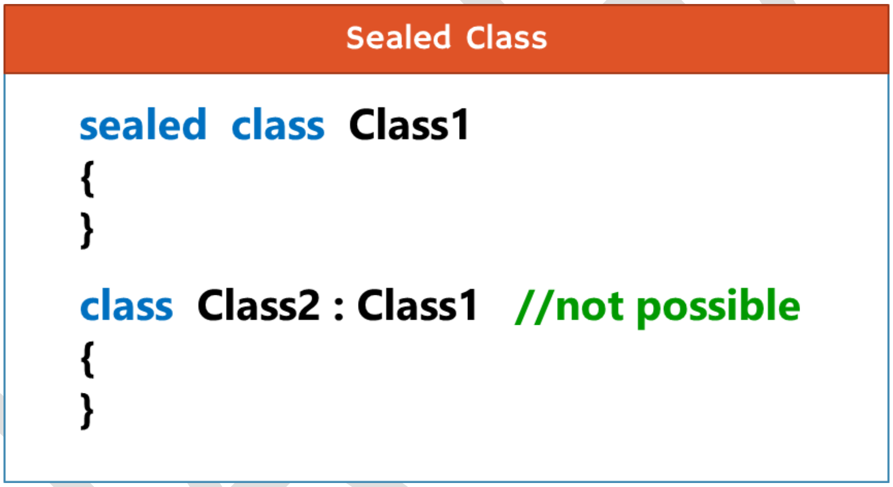
16

Sealed Classes

Introducing Sealed Classes

- Sealed class is a class, which is instantiable; but not inheritable.
- Use sealed class, whenever you don't want to let other developers to create child classes for the specific class.



Sealed Class

```
sealed class Class1
{
}

class Class2 : Class1   //not possible
{
}
```

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
Sealed Class	Yes	Yes	No	Yes

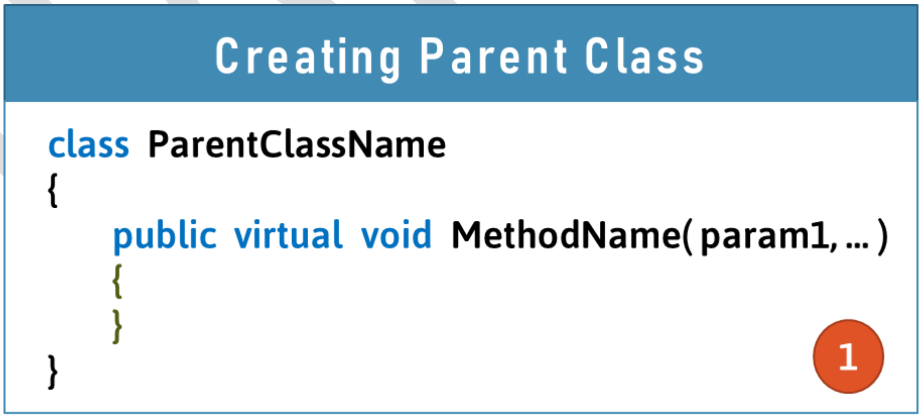
Class (vs) Sealed Class

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 Yes	Yes Yes	Yes Yes	Yes Yes
Abstract Class	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sealed							

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
Sealed Class	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes

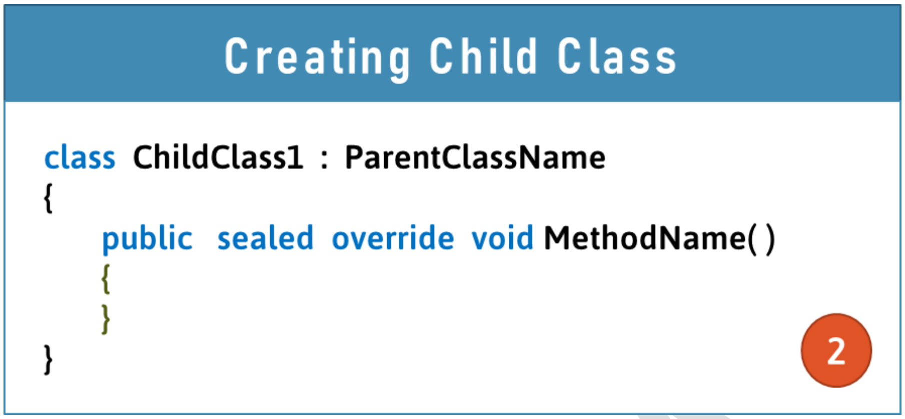
Sealed Methods

- Sealed Methods must be "override methods"; which can't be overridden in the corresponding child classes.
- Use sealed methods to prevent overriding that particular methods in the corresponding child classes.



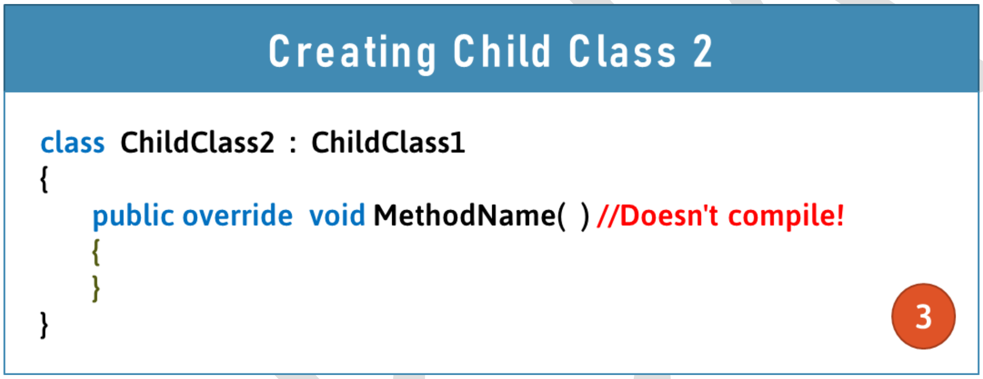
Creating Parent Class

```
class ParentClassName
{
     public virtual void MethodName( param1, ... )
     {
     }
}
```



Creating Child Class

```
class  ChildClass1  :  ParentClassName
{
     public  sealed  override  void MethodName( )
     {
     }
}
```



Creating Child Class 2

```
class  ChildClass2  :  ChildClass1
{
     public override  void MethodName(  ) //Doesn't compile!
     {
     }
}
```