

## Question

[Ken Earl](#) · Jul 25, 2020

## Abstract Classes and Methods

I have been looking into using abstract classes and methods. In other languages, if a superclass is defined as abstract, subclasses inheriting from the superclass will not compile until all abstract methods have been implemented in the subclass.

I have created the following classes to try this in Cache:

```
Class User.Music.AbstractClass [ Abstract ]
{

    Method testabs(inp As %String) [ Abstract ]{}

}

Class User.Music.AbstractTest Extends (User.Music.AbstractClass, %Persistent)

{

    Method testabs(inp1 As %Integer) { w !!,"Input:",$g(inp,"undefined") }

}
```

When I compile User.Music.AbstractTest it errors because the datatype of parameter inp1 doesn't match the superclass. On changing the parameter to %String it compiles ok (this makes sense to me).

However, if I comment out the implemented method testabs() from AbstractTest it compiles successfully and the object can be instantiated. But according to my understanding, now that testabs() has not been implemented it should either error in compile or when instantiated.

Also, if I add another parameter in addition to the parameters listed, this also seems to be ok.

Have I misunderstood how abstracted classes/methods work? How do others use abstracted classes/methods?

Many thanks.

[#Coding Guidelines](#) [#Caché](#)

---

Source URL:<https://community.intersystems.com/post/abstract-classes-and-methods>