

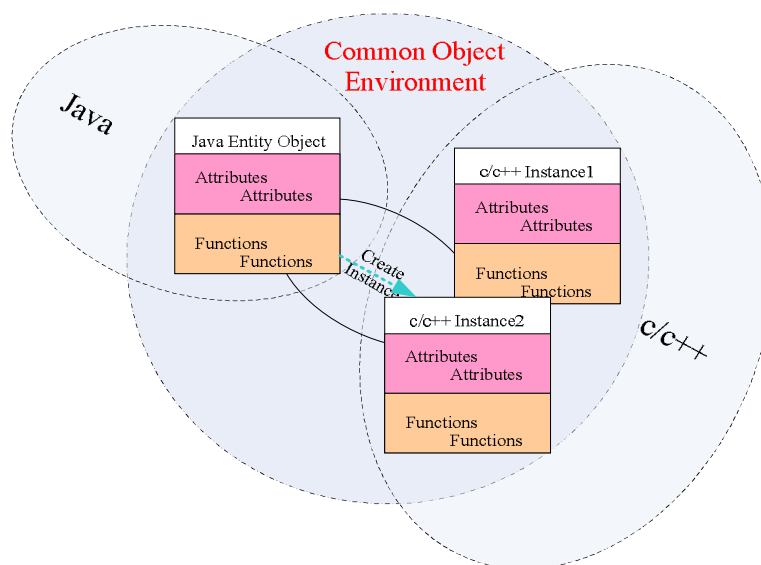


# **Multi-Language Programming : Entity Object Inheritance And Function Overloading**

For multi-language programming based on common object environment, object is defined by different languages. Therefore traditional object-oriented programming method is no longer suitable for this case. Developers can not create instance of a class defined in other language. For example, if a class is defined in java, a c++ instance of it can not be created.

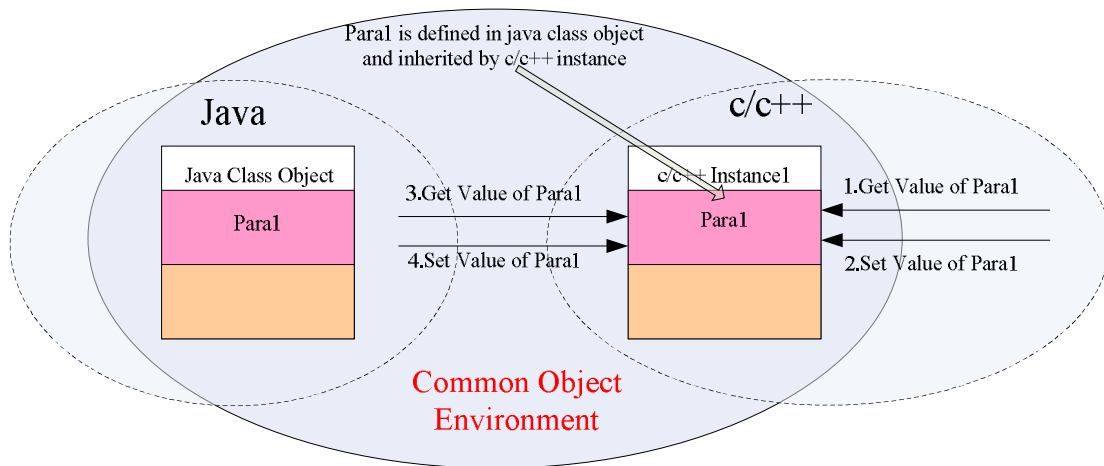
Interface objects managed in common object environment are entity objects rather than abstract definition of specific languages. They exist in memory of application. Object-oriented concept is very meaningful. It can simplify software development and maintenance. Is there a way to introduce the concept of object-oriented to these entity objects? How to handle entity objects inheritance and function overloading?

Based on the support of the environment, developers can create instances of an entity object (interface object). Example is as follows:

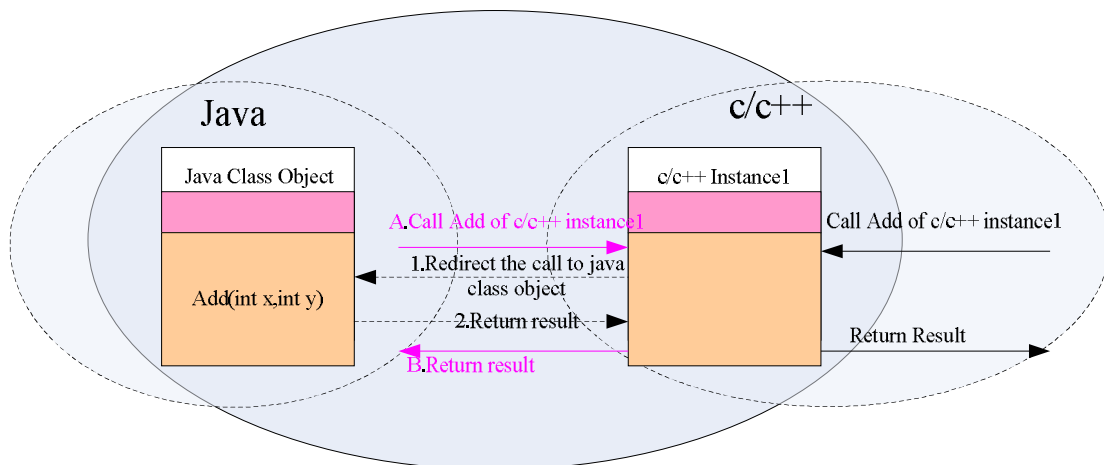


From above figure, we can see that entity objects can also act as class objects. They may have instances of other languages based on the support of common object environment. The instances can also define new attributes, new functions, or overloading functions defined in its class objects. Environment provides functions for application to access attributes or call functions of instances.

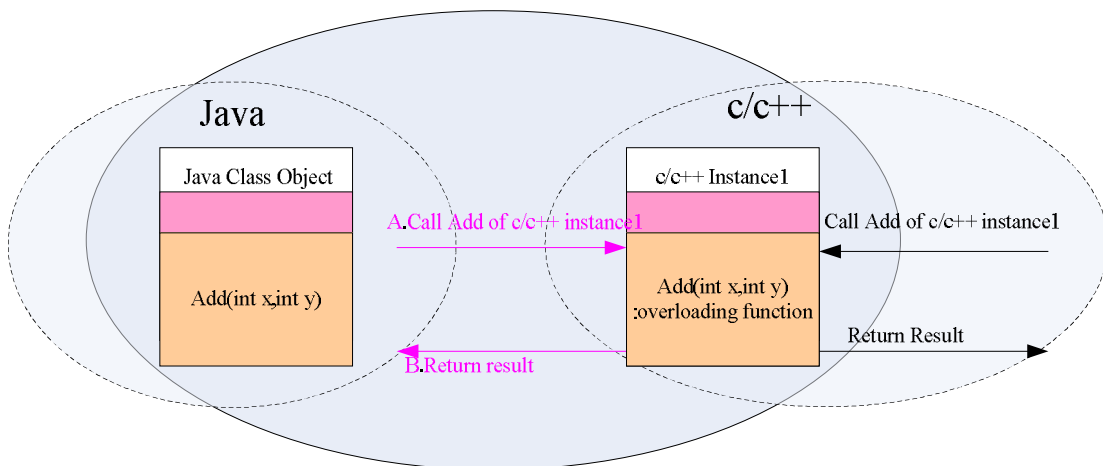
Attributes of objects may be defined by class object of other languages. The environment should provide functions for each language supported to access these attributes and perform parameter conversion. For examples, if class object of java defines an attribute Para1, its c/c++ instance will inherit the attribute Para1. Both languages can access Para1, which is shown below:



Calling objects functions is similar to accessing their attributes, common object environment supports to direct the call to correct entity object. For example, if a function “Add” is defined for java class object, calling the function of a c/c++ instance of it should be redirected to java language, which is shown below:



Function of class object may be overloading. In this case, calling of base function should be redirected to the overloading function. Common object environment will assist in the completion of the calling.



Traditional object-oriented method is supported by the compiler and language syntax. For entity objects in common object environment, because objects may be created and their functions may be defined by different languages, the language syntax and compiler can no longer support. To solve this problem, entity class objects are introduced and the environment runtime supports their inheritance and functions calling and overloading.

Further calling methods of objects belong to different languages will be discussed later.