

Lab Sheet 7

Understanding the Concept of Virtual function, Virtual base class and RTTI

Virtual Function

The overridden function in the derived class can be invoked by means of a base class pointer if the function is declared virtual in the base class. Suppose a virtual function `get()` is defined in the base class `Base` and again it is defined in the derived class `Derived`.

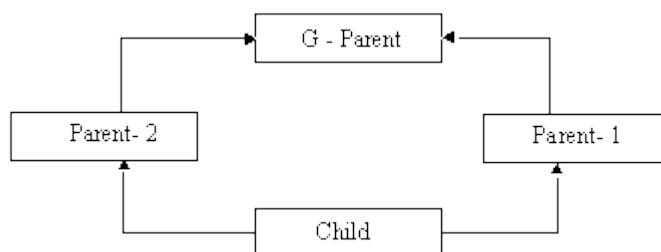
We can use the base class pointer to invoke the `get()` function of the derived class.

```
Derived d;  
  
Base *b;  
  
b=&d;  
  
b-> get ( )           //it calls the get ( ) function of the derived  
class.
```

Virtual Destructors

When a base class pointer that is pointing to a derived class object is deleted, destructor of the derived class as well as destructors of all its base classes is invoked, if the destructor in the base class is declared as virtual.

Virtual Base Class



In this type of inheritance there may be ambiguity in the members of the derived class `child` because it is derived from two base classes, which are again derived from the same base class. Hence to avoid this ambiguity the class `G – parent` can be made virtual.

Runtime Type Information (RTTI)

The runtime type information is one of the features of C++ that exhibit runtime polymorphic behavior. In C++ we can find the type information of an object at runtime and change the type of the object at runtime. The operators `dynamic_cast` and `typeid` are used for runtime type information.

For example if `Animal` is a polymorphic base class and `Dog` and `Cat` are derived classes of base class `Animal` then

```
Animal *anmp;  
Dog dg;  
Cat ct;  
anmp=&dg;
```

```
cout<< typeid(*anmp).name();
```

displays the information of the object pointed by `anm` pointer

Similarly

```
Cat *cpt;  
cpt=dynamic_cast<Cat*>(panm);
```

The down cast is successful if `panm` is holding the address of objects of class `Cat`.

Exercises

1. Write a program to create a class **shape** with functions to find area of the shapes and display the name of the shape and other essential component of the class. Create derived classes circle, rectangle and trapezoid each having overridden functions area and display. Write a suitable program to illustrate virtual functions and virtual destructor.
2. Create a class **Person** and two derived classes **Employee**, and **Student**, inherited from class **Person**. Now create a class **Manager** which is derived from two base classes **Employee** and **Student**. Show the use of the virtual base class.
3. Write a program with **Student** as abstract class and create derive classes **Engineering**, **Medicine** and **Science** from base class **Student**. Create the objects of the derived classes and process them and access them using array of pointer of type base class **Student**.
4. Create a polymorphic class **Vehicle** and create other derived classes **Bus**, **Car** and **Bike** from **Vehicle**. With this program illustrate RTTI by the use of `dynamic_cast` and `typeid` operators.