

Agenda

- Upcasting
- Downcasting
- Object slicing
- Function overriding
- Early binding and Late binding.
- Virtual function
- Pure virtual function
- abstract class

upcasting.(demo01)

- Base class pointer or reference pointing at derived class object is called as upcasting.

Object slicing (demo02)

- When we keep object of derived class into base class pointer or reference then the base class reference or pointer can only point at the base class members inherited into derived class.
- They cannot point at the members of the derived class.
- this is what we call it as object slicing.

DownCasting (demo03)

- Converting/Assigning base class pointer into derived class pointer is called as downcasting.
- at the time of downcasting you must be aware that the base class pointer is pointing at an derived class object
- It means upcasting must be done before downcasting

Function overriding (demo04)

- When your derived class redefines the same function from base class that is inherited with same name and signature then we call it as function overriding.
- function overriding is an example of run time polymorphism.

Virtual function,(Early binding and Late binding) (demo04 to demo06)

- When you override the functions inside derived class then even with the base class pointer when such overridden functions are called the base class functions get called.

- this is because of early binding, here the function that are getting called are finalized at compile time.
- If you want to call the derived versions of the functions on the base class pointer or reference then we have to do late binding.
- to have late binding declare the functions as virtual inside the base class.

Pure virtual functions (demo07)

- If the implementation of the functions is not known inside the base class and we want the derived classes to implement these functions, make such functions as pure virtual inside base class.
- to make the functions as pure virtual declare the functions as virtual and assign 0 value to it.

Abstract Class (demo07)

- The class in which pure virtual functions exist is called as abstract class.
- we cannot create object of the abstract class.
- we can create only pointer or reference of the abstract class.

Inheritance examples using virtual functions (demo08)

example of enum and menu driven code using run time polymorphism (demo09 and demo10)

interface implementation example using pure virtual functions (demo11)

Multiple file demo example using demo11 (demo12)

abstract implementation example (demo13)

Adding Association inside inheritance (demo14)

Creating an array of base class pointers (demo 15)
