

Agenda

- Revision
- Object slicing
- upcasting and down casting.
- Function overriding
- Virtual Function
- Early binding and late binding
- virtual destructor
- Virtual Function Table and Virtual Function Pointer
- Pure virtual function
- abstract class and interface.
- frined class

Upcasting,Downcasting,Object Slicing (demo01 to demo03)

- Assigning the object of derived class into the base class reference or pointer is called as upcasting.
- When upcasting is done the base class pointer or reference can point to only the base members that exists inside the derived class.
- It cannot point to the members of the derived class even if we have created the object of derived class.
- this is called as object slicing.
- to access the members of the derived class using base class reference or pointer, we have to convert the base class pointer/reference to the derived class pointer/reference
- the conversion of base class pointer/reference into the derived class pointer/reference is called as downcasting.
- at the time of downcasting explicit typecasting is mandatory.

Function Overiding, Early binding (demo04)

- Defining the same function of base class into the derived class with same name and same signature is called as function overriding.
- When the base class pointer/reference pointing at the derived class object calls such function, then always the base class function gets executed.
- this is because of ealy binding.
- the function call that binds at compile time is called as early binding.

Late Binding, Virtual Function (demo05)

- If we want to call the overridden function from the derived class using base class pointer/reference then we have to perform late binding.
- when the functions gets bind at run time it is called as late binding.
- To perform late binding the function inside the base class should be made as virtual.

virtual destructor (demo06)

- When the pointer is of base class and object is of derived class then at time of deallocating the dynamic memory of derived object using base class ptr, only the base class dtor get called.
- to call the derived class dtor and then the base class dtor declare the dtor of the base class as virtual.

virtual pointer, vtable

- It is a pointer that a compiler adds up in the Base class where the virtual functions are declared.
- It points to the vtable of that class object.
- The vtable consists of the address to the virtual functions of that class.

pure virtual function , abstract class (demo07)

- if the implementation of the functions is not known inside the base class and the base class wants the derived classes to implement the functions with same design, declare such functions as pure virtual inside the base class.
- to declare the functions as pure virtual, declare them as virtual and assign it with 0.
- the class in which pure virtual function exists, these classes become the abstract class.
- we cannot create object of the abstract class.
- however we can create the pointer or reference of the abstract class.
- If the derived classes do not override the pure virtual functions inherited from base class then the derived classes will also become the abstract class.

abstract implementation (demo08 to demo10)

interface implementation (demo11)

array in inheritance (demo12)

friend class (demo13)

- it is a class which is able to access the private members of the class in which it is declared as a friend.
- Generally we can declare the derived classes as friend inside the base class.

LabWork

- Complete the demos 1 by 1 and parallelly read the slides.
- once all the demos are done then have the implementation of abstract and interface on your own.
- Solve the given assignment.
- preferred using enum in menu driven code even not used its ok.