Object Oriented Programming Using C++

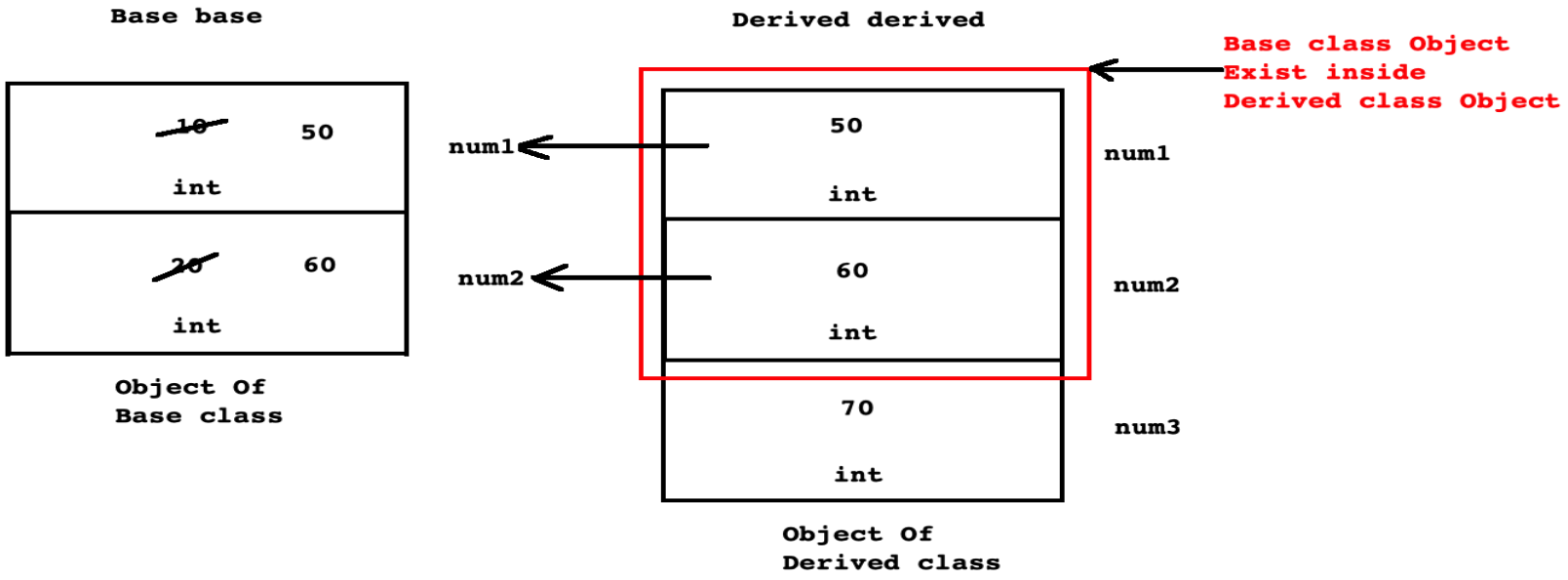
Ketan G Kore
ketan.kore@sunbeaminfo.com

Inheritance Basics - Revision

- During inheritance, members(data member, member function, nested types) of derived class do not inherit into base class. Hence using base class object, we can access members of base class only.
- During inheritance, members(data member, member function, nested types) of Base class inherit into derived class. Hence using derived class object, we can access members of base class as well as derived class.
- Members of base class inherit into derived class. Hence we can consider object of derived class as a object of base class.
- Since derived class object can be considered as base class object, we can use it in place of base class object.

Object Slicing

- If we assign, derived class object to the base class object then compiler consider only base class portion from derived class object and copy it into base class object. This process is called object slicing.
- **During object slicing, node of inheritance must be public.**



[Object Slicing]

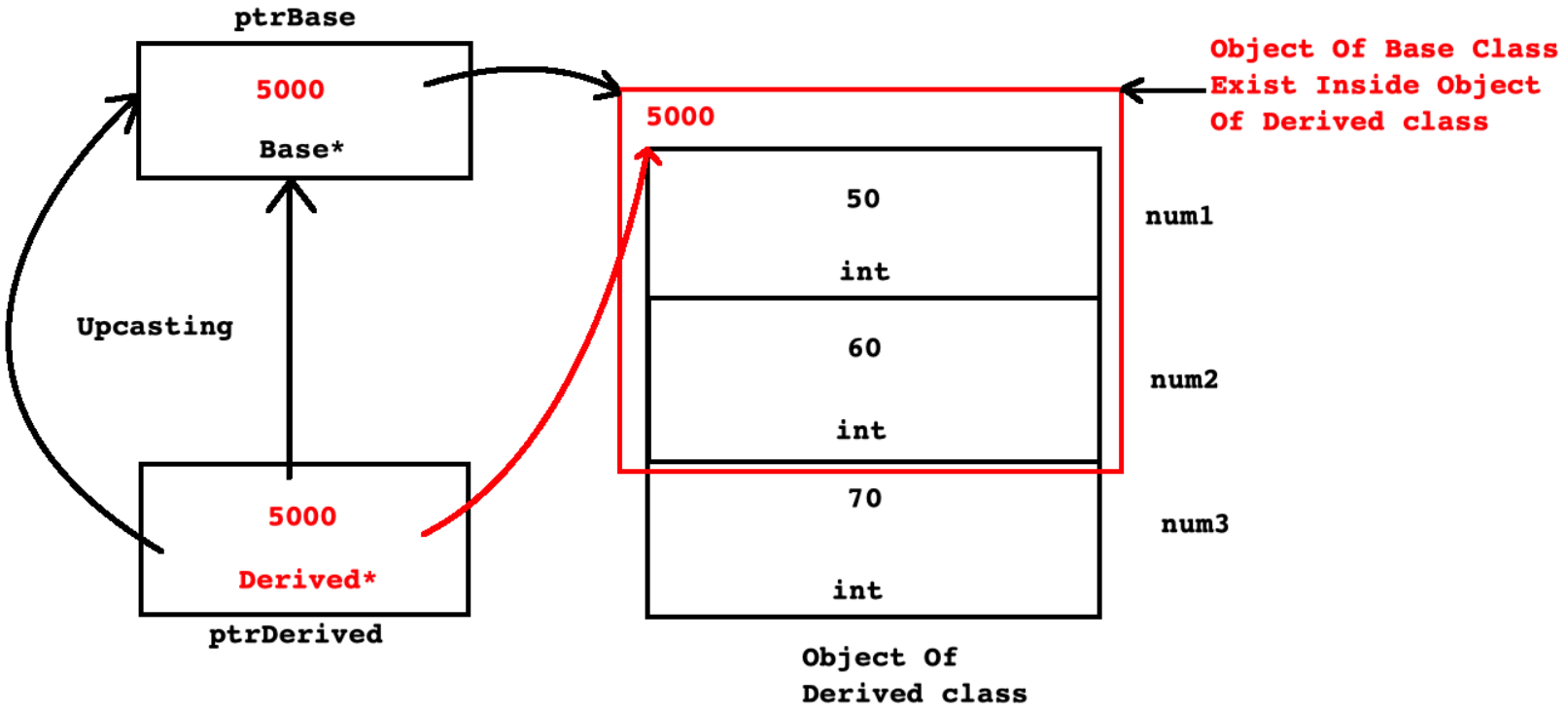
Upcasting

- Process of converting, pointer of derived class into pointer of base class is called upcasting.

```
Derived derived(50, 60, 70);
Derived *ptrDerived = &derived;
//ptrDerived->printRecord( ); //Derived::printRecord
//Base *ptrBase = ( Base* )ptrDerived;    //Upcasting
Base *ptrBase = ptrDerived;    //Upcasting
//ptrBase->printRecord( ); //Base::printRecord
```

- We can store address of derived class object into pointer of base class. It is also called as upcasting.

Upcasting



- We can convert, pointer of derived class into pointer of base class. It is called upcasting.
- Using upcasting, we can reduce object dependancy in the code. Which helps developer to reduce maintenance of the code.
- Here using ptrBase we can access:
  1. num1 and num2
  2. showRecord and printRecord of base class.
- Here using ptrDerived we can access:
  1. num1, num2 and num3
  2. showRecord, printRecord as well as displayRecord

Down casting

- Process of converting, pointer of base class into pointer of derived class is called down casting.

```
Base *ptrBase = new Derived( 50, 60, 70 );   //Upcasting
ptrBase->printRecord( );     //Base::printRecord
Derived *ptrDerived = ( Derived* )ptrBase;   //Downcasting
ptrDerived->printRecord( );     //Derived::printRecord
```

- During upcasting, if we want to access:
    1. Data members of derived class
    2. Non overridden function of derived class

  then we should use down casting.

Thank you