Program of Base Class Pointer and Derived Class Object

```
#include<iostream.h>
#include<conio.h>

class basic
{

 public:

  basic()
  {
   cout<<"Constructer of basic called\n";
  }

  ~basic()
  {
    cout<<"Destructer of basic called\n";
  }

  void fun1()
  {
    cout<<"fun1 called\n";
  }

  void fun2()
  {
   cout<<"fun2 called\n";
  }
};

class advance : public basic
{


 public:

  advance()
  {
   cout<<"Constructer of advance called\n";
  }

  ~advance()
  {
    cout<<"Destructer of advance called\n";
  }
  void fun3()
  {
    cout<<"fun3 called\n";
  }
  void fun4()
  {
   cout<<"fun4 called\n";
  }
};

void main()
{
 clrscr();

 advance a;
 basic *ptr;
 ptr=&a;
```

```
ptr->fun1();
ptr->fun2();
// ptr->fun3();
// ptr->fun4();
// delete ptr;

/*
 //Aliter
 basic *ptr;
 ptr=new advance();

 ptr->fun1();
 ptr->fun2();   */
// ptr->fun3();
// ptr->fun4();
// delete ptr;

 getch();
}
```

Note:

Fun3() and fun4() can not be called because they are available in advance class, and pointer is made of basic class only.