[Skip to main content](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/courseware/a71234edfd4f465e8d3e7e0082d04540/d8547e00461e47cbbdaf79f163ef627f/1?activate_block_id=block-v1%3AIITBombayX%2BCS101.2x%2B2022_T1%2Btype%40vertical%2Bblock%40a02477413255410f8c70744f99869ac9#main)

**[IITBombayX Home Page](http://iitbombayx.in/)**

**IITBombayX: CS101.2xObject-Oriented Programming**

[Courses](http://iitbombayx.in/courses)

[Help](https://iitbombayx.in/contact_us)

[[https://courses.iitbombayx.in/static/images/profiles/default_50.3455a6581573.png](https://courses.iitbombayx.in/dashboard) Dashboard for:Paramhans\_Singh](https://courses.iitbombayx.in/dashboard)

1. [Course, current location](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/course/)

1. [Discussion](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/discussion/forum/)

1. [Progress](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/progress)

1. [Grading Policy](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/487375f5363a49a492641b63d5e814ec/)

1. [Course Slides](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/pdfbook/0/)

1. [Practice Programming](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/b5c2b3240ef9416ebb0ea5b2087cb4ee/)

1. [FAQs](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/f2e9c32b35ae4ef083ed716b4981ff0d/)

[Course](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/course/)  [Topic 2: Review of Pointers and Structures](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/course/#block-v1:IITBombayX+CS101.2x+2022_T1+type@chapter+block@a71234edfd4f465e8d3e7e0082d04540)  [Session 10: Structures and Pointers](https://courses.iitbombayx.in/courses/course-v1:IITBombayX+CS101.2x+2022_T1/course/#block-v1:IITBombayX+CS101.2x+2022_T1+type@sequential+block@d8547e00461e47cbbdaf79f163ef627f)  Walkthrough

 PreviousNext

1. video Structures and Pointers
2. other Slides: Structures and Pointers
3. other Walkthrough

**Walkthrough**

 Bookmark this page

**this Pointer**

**Objective:**

To learn about a special type of pointer known as **this**pointer. We will discuss what is **this**pointer, how it is used, its applications and benefits.

**Let's Start:**

In C++, a copy of each non-static member variable is kept for each of the objects of that Class. That's why we can keep non-static variables of each object at a different state. But the member functions of a Class are not stored on per object basis. They are stored as one single copy for all the class just like static variables. But as we know we can invoke member functions through objects as **object.member\_function(args)** , How does C++ keep track of which object has called the function?

The answer is **this**pointer.

In C++, each class object has a pointer (defined implicitly) that holds its address. This special type of pointer is known as **this** pointer. This pointer is passed implicitly to every member function of the class as an argument ( except static functions as they don't need an object to be called).  So any member function has a hidden argument of **this**pointer. Let's see an example.

class A

{

   public:

   int x;

   void assign(int y)

   {

      x = y;

   }

};

A object;

object.assign(5);

Now, how does the assign function needs to know which object has called it so as to update the corresponding **x**? This problem is solved using an implicit pointer **this**pointer.  The assign function is actually defined in the following way:

void assign(A const\* this, int y);

This first parameter remains hidden but is always present in case of non static functions. So the call to the assign function is implicitly interpreted as :

object.assign(&object, 5);

**Uses :**

There are many uses of **this**pointer:

1. **Resolve the identity of class variables.**

One use as we discussed above is to resolve the identity of the class variables inside member functions which might be shadowed by local variables defined inside. One more example is given below:

class A

{

   public:

   int x;

   A(int x)

   {

      this->x = x;

   }

   bool isGreater(A object)

      {

              if (this->x > object->x)

      {

           return True;

      }

      return False;

}

2.**Returning reference of the calling object.**

Suppose we want to write a function which returns the reference of the calling object. We can do so using the **this**pointer as:

A& function()

{

      // rest of the code

      return \*this;

}

3. **Calling member functions in chains**

**this**pointer can also be used for calling member functions in chains. An example is given below:

class A

{

   int X;

   int Y;

   A& setX(int a)

  {

      X = a;

     return \*this;

  }

   A& setY(int a)

  {

      Y = a;

     return \*this;

  }

};

A object;

object.setX(5).setY(10);

Although chaining can be implemented when the functions return any object pointer but if you want to return the pointer pointing to the calling object, you can use **this** pointer as shown above.

**CONCLUSION:**

You should play more with **this**pointer and explore more about it's uses. **this**pointer is frequently used in the standard Object Oriented Programming problems and it's necessary that you should be comfortable with using them.

 PreviousNext

© All Rights Reserved

[IITBombayX Home Page](http://iitbombayx.in/)

1. [About](http://iitbombayx.in/about)
2. [Research](http://iitbombayx.in/research)
3. [FAQs](http://iitbombayx.in/faq)
4. [Contact Us](http://iitbombayx.in/contact_us)

* [MOOCs Copyright](http://iitbombayx.in/mooc-copyright-guidelines)
* [Terms of Service and Honour Code](http://iitbombayx.in/honor)
* [Authorization Document](http://iitbombayx.in/authorization)
* [Privacy Policy](http://iitbombayx.in/privacy)

[Powered by Drupal](https://www.drupal.org/)

[Creative Common License](https://creativecommons.org/)

[](http://open.edx.org/)