6/8/2011

The Pointer ‘this’ in C++

The mystical powers of the ‘this’ pointer in C++.

dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Definition

- Pointer accessible only within the non-static member functions of a **class**, **struct**, or **union** type.
- It points to the object for which a member function is called.
- Its use is legal but not necessary.

```
class Date
{
    int day;
    int month;
    int year;
    ...
}
```

```
void SetMonth( int mn )
{
    month = mn;              // These three statements
    this->month = mn;        // are equivalent
    (*this).month = mn;
}
```

Uses

- Eliminates ambiguity in the source code. Ambiguity is usually caused by **lack of creativity**.

```
void SetDate(int day, int month, int someYear)
{
    this->day = day;
    this->month = month;
    this->year = someYear;
}
```

- The address of the object is passed to the function implicitly when called (i.e. as a hidden argument). As an example, the following function call:

```
myDate.setMonth( 3 );

setMonth( &myDate, 3 ); //what the compiler sees
```

Uses

- Occasionally, it is used **explicitly** (i.e. to manipulate self-referential data structures, where the address of the current object is required).

```
Date::NextYearsDate(){
    Date td = FunctionNotInDateClass::CopyDate(this);
    td->year++;
}
```

- Also used to guard against self-reference

```
if (&Object != this) {
    // do not execute in cases of self-reference
    ...
```

Examples

```
int NotSoObviousFunction(int day)
{
    return day;
}
```

```
int CompareDatesAmbiguously(Date someDate)
{
    int day = someDate.day;
    int month = someDate.month;
    int year = someDate.year;

    if(day > day) return 1;
    else if(day==day) return 0;
    else return -1;
}
```

Non-Ambiguous Implementation

```
int UnambiguousFunction(int day)
{
    return this->day;
}
```

```
int CompareDates(Date someDate)
{
    int day = someDate.day;
    int month = someDate.month;
    int year = someDate.year;

    if(this->day > day) return 1;
    else if(this->day == day) return 0;
    else return -1;
}
```

Interesting Facts

The **this** pointer is:

- not counted for calculating the size of the object.
- not accessible for static member functions.
- not modifiable (i.e. this = someValue is not allowed).
- not NULL.

In contrast to C++, Java's **this** is a reference.

Q&A

Resources

- http://msdn.microsoft.com/en-us/library/y0dddwwd(v=vs.80).aspx
- http://www.codersource.net/c/c-tutorials/c-tutorial-this-pointer.aspx
- http://login2win.blogspot.com/2008/05/c-this-pointer.html
- http://bytes.com/topic/c/answers/63685-differences-between-c-java-pointer
- http://promoimg.beckett.com/news/news-content/uploads/2011/05/question.jpg