“THIS” POINTER

Sukhbir Tatla
sukhbir.tatla@senecacollege.ca

INTRODUCTION

- Every object in C++ has access to its own address through an important pointer called this pointer.
- The ‘this’ pointer is passed as a hidden argument to all non-static member function calls and is available as a local variable within the body of all non-static functions.
- ‘this’ pointer is not available in static member functions as static member functions can be called without any object (with class name).
- Friend functions do not have a this pointer, because friends are not members of a class. Only member functions have a this pointer.

USING ‘this’ POINTER

- **`this`** pointer
  - Allows objects to access their own address
  - Not part of the object itself
  - Implicit first argument on non-**`static`** member function call to the object
  - Implicitly reference member data and functions
- Example: class **`Employee`**
  - For non-**`const`** member functions: type **`Employee * const`**
    - Constant pointer to an **`Employee`** object
  - For **`const`** member functions: type **`const Employee * const`**
    - Constant pointer to an constant **`Employee`** object

USING ‘this’ POINTER

- Cascaded member function calls
  - *Function returns a reference pointer to the same object*

    { return *this; }
  - *Other functions can operate on that pointer*
  - *Functions that do not return references must be called last*

Program Example

```
#include <iostream>
using namespace std;
class Box {
private:
        double length, width, height;    // Dimensions of a box
public:
// Constructor definition
Box (double l = 2.0, double b = 2.0, double h = 2.0)         {
        cout << "Constructor called." << endl;
        length = l;      width = b;      height = h;      }
double Volume()      {
                return length * width * height;    }
int compare(Box box)      {
                return this->Volume() > box.Volume();     }
};      //Class Ends
```

Program Example

```
int main() {
        Box Box1(3.3, 1.2, 1.5);    // Declare box1
        Box Box2(8.5, 6.0, 2.0);    // Declare box2
        if (Box1.compare(Box2)) {
                cout << "Box2 is smaller than Box1" << endl;
        }
        else {
                cout << "Box2 is equal to or larger than Box1" << endl;
        }
        cout << endl;
        return 0;
}       //Main Ends
```

When local variable’s name is same as member’s name

```
#include<iostream>
using namespace std;
/* local variable is same as a member's name */
class Test                   {
private:  int x;
public:
       void setX (int x)     {
       // The 'this' pointer is used to retrieve the object’s x hidden by the local variable ‘x’
            this->x = x;     }
            void print()     {          cout << "x = " << x << endl;          }
            };
```

When local variable’s name is same as member’s name

```
int main()
{
        Test obj;
        int x = 20;
        obj.setX(x);
        obj.print();
        return 0;
}       // Main Ends
```

Solve the Following:

- Predict the Output or Compilation Error for the following programs in the next slides.
- If there is/are compilation error(s), solve the error(s) and execute the program to check the output.

Question 1:

```
#include<iostream>
using namespace std;
class Test        {
private:
        int x;
public:
        Test(int x = 0) { this->x = x; }
        void change(Test *t) { this = t; }
        void print() { cout << "x = " << x
<< endl; }
};      //Class Ends
```

```
int main()        {
        Test obj(5);
        Test *ptr = new Test (10);
        obj.change(ptr);
        obj.print();
        return 0;
}       // Main Ends
```

Question 2:

```
#include<iostream>

using namespace std;

class Test        {

private:

  int x;

  int y;

public:

  Test(int x = 0, int y = 0) { this->x = x;
this->y = y;         }

  static void fun1() { cout << "Inside
fun1()";             }

  static void fun2() { cout << "Inside
fun2()"; this->fun1();       }

};          //Class Ends
```

```
int main()

{

            Test obj;

            obj.fun2();

            return 0;

}           // Main Ends
```

Question 3:

```
#include<iostream>

using namespace std;

class Test          {

private:

            int x;

            int y;

public:

  Test (int x = 0, int y = 0) { this->x = x;
this->y = y;        }

  Test setX(int a) { x = a; return *this; }

  Test setY(int b) { y = b; return *this; }

  void print() { cout << "x = " << x << " y =
" << y << endl;   }

};          //Class Ends
```

```
int main()

{

            Test obj1;

            obj1.setX(10).setY(20);

            obj1.print();

            return 0;

}           // Main Ends
```

Question 4:

```
#include<iostream>

using namespace std;

class Test          {

private:

  int x;

  int y;

public:

  Test(int x = 0, int y = 0) { this->x = x;
this->y = y;          }

  void setX(int a) { x = a; }

  void setY(int b) { y = b; }

  void destroy()  { delete this; }

  void print() { cout << "x = " << x << " y =
" << y << endl; }

};          //Class Ends
```

```
int main()

{

  Test obj;

  obj.destroy();

  obj.print();

  return 0;

}          // Main Ends
```

THANK YOU!

Any Questions Please?