

# Material 05 Lab - Comparison of C++ and Java

---

## Table of Contents

- [Take Home](#)
  - [Homework - Lab 02](#)
  - [Homework - Lab 03](#)
  - [Homework - Lab 05](#)

## Take Home

### Homework - Lab 02

在 Java 中可以使用 `super.super.method()` 這樣的語法嗎（請試試看）？

```
// Lab02.java

class Lab02 {
    public static void main(String[] args) {

        // Base class
        Base base = new Base();
        base.baseMethod();

        // FirstSubClass
        FirstSubClass firstSubClass = new FirstSubClass();
        SecondSubClass secondSubClass = new SecondSubClass();
        firstSubClass.FirstSubClassMethod();
        secondSubClass.SecondSubClassMethod();

    }
}

class Base {
    Base() {}
    Base(int w) {}
    Base(int w, String f) {}

    void baseMethod() {
        System.out.println("Base method");
    }
}

class FirstSubClass extends Base {
    FirstSubClass() {}

    FirstSubClass(int w) {
        super(w);
    }
}
```

```
    FirstSubClass(int w, String f) {
        super(w, f);
    }

    void FirstSubClassMethod() {
        super.baseMethod();
    }
}

class SecondSubClass extends Base {
    SecondSubClass() {}

    SecondSubClass(int w) {
        super(w);
    }

    SecondSubClass(int w, String f) {
        super(w, f);
    }

    void SecondSubClassMethod() {
        // super.super.baseMethod();    // Wrong
        super.baseMethod();             // Correct
    }
}
```

透過這個命題，我撰寫了一個簡單的 JAVA 程式碼範例來驗證，答案是不行使用 `super.super.method()` 這樣的語法，會出現以下的錯誤訊息。

```
Syntax error on token "super", Identifier expected
```

我們如果要呼叫父類別的父類別的方法，直接使用 `super.method()` 即可。

## Homework - Lab 03

請分別舉例描述 C++ 與 JAVA 如何從子類別建構子傳遞參數到父類別建構子。

```
#include <iostream>

class Parent {
public:
    Parent(int x) {
        std::cout << "Parent constructor called with parameter: " << x <<
std::endl;
    }
};

class Child : public Parent {
public:
```

```
Child(int y) : Parent(y) {
    std::cout << "Child constructor called with parameter: " << y <<
std::endl;
}
};

int main() {
    Child childObj(10); // 建立 Child 物件時傳遞參數到父類別建構子
    return 0;
}
```

在這個 C++ 程式中，使用了 pass by value 的方式來將參數傳遞到父類別建構子中。

在 Child 的建構子中的初始化列表：`Parent(y)` 是使用 pass by value 的方式，因為它將整數參數 `y` 直接傳遞給 Parent 的建構子 `Parent(int x)`。這裡的 `y` 是以值的形式傳遞給 Parent 的建構子，而不是以 pointer 或 reference 的形式。

```
public class Lab03 {
    public static void main(String[] args) {
        Child childObj = new Child(10); // 建立 Child 物件時傳遞參數到父類別建構
子
    }
}

class Parent {
    Parent(int x) {
        System.out.println("Parent constructor called with parameter: " +
x);
    }
}

class Child extends Parent {
    Child(int y) {
        super(y); // Call parent constructor with parameter
        System.out.println("Child constructor called with parameter: " +
y);
    }
}
```

在以上的 JAVA 程式中，使用了 pass by value 的方式來將參數傳遞到父類別建構子中。另外如果要做到傳遞參數到父類別建構子，需要使用 `super()`。

## Homework - Lab 05

請描述 C++ 關鍵字 `virtual` 與 JAVA 關鍵字 `abstract` 的異同。

C++:

```
Class human {  
    int weight;  
    int height;  
Public:  
    virtual void walk() = 0;  
    virtual void speak() = 0;  
}
```

Java:

```
public abstract Class human {  
    int weight ;  
    int height ;  
    public abstract void walk() ;  
    Public abstract speak();  
}
```

用上課範例中，當我們今天要在 C++ 時做出 Abstract 時，我們會使用 **virtual** 並且在後面加上 **= 0** 來表示這是一個 Abstract Function。而在 JAVA 中，我們會使用 **abstract** 來表示這是一個 Abstract Class 或是 Abstract Function，另外在 Java 中，如果一個 Class 中，只要一個 method 是 **abstract**，那這個 Class 就必須要是 **abstract**。