



Padanan C++ – Java

IF2210 – Semester II 2024/2025

by: RSP & IL; rev: SAR

C++

Java

1. **cctor**
2. **ctor initialization list**
3. **Operator overloading**
4. Parameter passing: by value, by ref
5. Templates
6. class
7. Inheritance: public, protected, private
8. Multi-inheritance
9. **ABC**
10. **Abstract class**
11. Pemanggilan ctor base class eksplisit tidak bisa

1. Tak ada terjemahan
2. Tak ada
3. **Tak ada, fungsi biasa**
4. Hanya by value
5. Kelas generik
6. public class, private class, protected, package private
7. Inheritance: hanya ada public
8. “tidak boleh” + interface
9. **Interface**
10. **Abstract class**
11. Ctor superclass bisa diinvokasi: `super()` – hanya baris pertama ctor

C++

Java

1. friend
2. main program tidak mungkin dalam kelas
3. Scope: kelas + file
4. namespace
5. bool
6. string
7. Try-catch: tidak ada keyword “finally”
8. Throw objek apapun, walau ada kelas “exception”

1. Tidak ada friend
2. main adalah *method* statik sebuah kelas
3. Scope: kelas (*)
4. package
5. Boolean
6. String
7. Try-catch: Ada keyword “finally”
8. Throw: harus objek turunan throwable

Class Modifier (Java)

Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

<https://docs.oracle.com/javase/tutorial/reflect/class/classModifiers.html>

<https://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>

The first data column indicates whether the class itself has access to the member defined by the access level. As you can see, a class always has access to its own members. The second column indicates whether classes in the same package as the class (regardless of their parentage) have access to the member. The third column indicates whether subclasses of the class declared outside this package have access to the member. The fourth column indicates whether all classes have access to the member. Access levels affect you in two ways. First, when you use classes that come from another source, such as the classes in the Java platform, access levels determine which members of those classes your own classes can use. Second, when you write a class, you need to decide what access level every member variable and every method in your class should have.

Catatan

- › Interface – hanya ada di Java:
<https://docs.oracle.com/javase/tutorial/java/concepts/interface.html>
- › Interface (Java) diterjemahkan menjadi ABC
- › Abstract class CPP menjadi abstract class Java
- › Instansiasi kelas generik menjadi kelas spesifik: di Java tidak dapat menggunakan type primitif. Contoh:

```
Stack<int> S;      //tidak mungkin di Java  
Stack<Integer> S; // boleh
```