មជ្ឈមណ្ឌលកូរ៉េ សហវែរ អេច អ ឌី

Korea Software HRD Center

កម្មវិធីបង្រៀន Java Programming ជាភាសាខ្មែរ

Online Java Training Course

Advisor: Dr. Kim Tae Kyung

Webcash

KOICA
Korea International Cooperation Agency

www.kshrd.com.kh

មេរៀនទី២៖ Overriding

២.១ . សេចក្តីផ្តើម

២.២. ការប្រើ Method Overriding

២.៣. ការប្រើ Super Keyword

២.១ . សេចក្តីផ្តើម

- ជាដំបូង មុននឹងចាប់ផ្តើមសិក្សាអំពី **Overriding** យើងត្រូវស្គាល់ SuperClass និង SubClass ជាមុនសិន ។
  - ☞ Super Class: គឺជា Class ដែលត្រូវបានគេធ្វើការ Inherit ។
  - ☞ Sub Class: គឺជា Class ដែលបានធ្វើការ Inherit ពី Super Class ។
- **Overriding:** គឺជាការទាញយក Method ទាំងឡាយណារបស់ Super Class ដែលយើងត្រូវការប្រើប្រាស់ ហើយបន្ថែម ឬកំណត់មុខងារ (Functionality)ថ្មី ទៅអោយ Method ទាំងនោះ ។
- សារៈប្រយោជន៍របស់ **Overriding** គឺផ្តល់លទ្ធភាពអាច ឲ្យយើងកំណត់នូវ Behavior ជាក់លាក់ទៅអោយ Sub Class មានន័យថា Sub Class មួយអាច Implement នូវ Method របស់ Super Class បាន តាមតំរូវការរបស់យើង ។

២.២. ការប្រើ Method Overriding

- ឈ្មោះ, **Argument list** និង **Return Type** ត្រូវតែដូចគ្នាទៅនឹង Overridden Method ។
- **Access Level** របស់ Method នៃ Sub Class ត្រូវតែមានកម្រិតស្មើ ឬធំជាង Access Level របស់ Overridden Method ។
- **Instance Method** អាច Override បានតែក្នុង Sub Class តែប៉ុណ្ណោះ ។
- បើ Method ណាដែលមិនអាចឲ្យគេ Inherit បាន វាក៏មិនអាចឲ្យគេ Override បានដែរ ។
- **Final Method** មិនអាច Override បានទេ ។
- **Static Method** មិនអាច Override បានទេ តែអាចប្រកាសម្តងទៀត ។

២.២. ការប្រើ Method Overriding

- **Sub Class** ដែលនៅក្នុង Package ជាមួយ Super Class របស់វា អាចធ្វើការ Override Method របស់ Super Class ដែលមិនមែនជា Private ឬ Final ។
- **SubClass** ដែលនៅក្នុង Package ផ្សេងគ្នាពី Super Class របស់វា អាចធ្វើការ Override ចំពោះ Method របស់ Super Class ណា ដែលជា Public ឬ Protected (Non-Final Methods) ។
- **Constructor** មិនអាចធ្វើឲ្យគេ Override បានទេ ។

២.៣. ការប្រើ Super Keyword

- **Super Keyword** ប្រើសំរាប់ធ្វើការ Access រាល់ Member របស់ Super Class។

```
//Animal.java
public class Animal {
        public void move(){
                System.out.println("Animals can move");
        }
}
//Dog.java
class Dog extends Animal{
        public void move(){
                super.move(); //Invokes the super class method
                System.out.println("Dogs can walk and run");
        }
}
//Test.java
public class Test {
        public static void main(String args[]){
                Animal b = new Dog(); // Animal reference but Dog object
                b.move(); //Runs the method in Dog class
        }
}
```

សមាជិក

ក្រុមអ្នកស្រាវជ្រាវ

ល. ខេង ចាន់វិជ្ជា
vichea@rocketmail.com

ល. សែម ចិត្រា
sabaychitra84@gmail.com

ក. លីម សុខហេង
lim.sokheng1@gmail.

ល. ឡៅ ស៊ុនឡេង
sunlenglao@gmail.com.com

ល. យ៉ូ វណ្ណារ៉ាវិទ្ធ
ravuthz@gmail.com

ក្រុមផលិតវីដេអូ

ល. ឈុន បញ្ញារតន៍
chhunpanharath@gmail.com

ល. ព្រាប វិទ្ធី
Itpreap.vuthy@gmail.com

ល. ហង្ស បូរី
houngboreyrupp@gmail.com

ល. ហួ ឈុនឡេង
huochhunleng@yahoo.com

មេរៀនបន្ទាប់ នឹងធ្វើការបង្ហាញពី Polymorphism