



មជ្ឈមណ្ឌលកូរ៉េ សហ្វ្វែរ អេច អ ឌី

Korea Software HRD Center

កម្មវិធីបង្រៀន Java Programming ជាកាសាខ្មែរ

Online Java Training Course

Advisor: Dr. Kim Tae Kyung



www.kshrd.com.kh

ជំពូកទី ៧

Common Class

មេរៀនទី១: Inheritance

មេរៀនទី២: Overriding

មេរៀនទី៣: Polymorphism

មេរៀនទី៣: Abstraction

មេរៀនទី៣: Encapsulation

មេរៀនទី៣: Interfaces

មេរៀនទី៣: www.Packages.kh

មេរៀនទី១៖ Inheritance

១.១. សេចក្តីផ្តើម

១.២. IS-A Relationship

១.៣. HAS-A Relationship

១.៤. The instance of Keyword

១.១ . សេចក្តីផ្តើម

□ **Inheritance:** អនុញ្ញាតិឱ្យ **Class** មួយអាចទទួលយក **Property** (**Fields, Methods**) នាំអស់ពី **Class** មួយទៀត យកមកប្រើប្រាស់ ដោយមិនចាំបាច់សរសេរ **Code** ឡើងវិញតាមរយៈការប្រើ **Keyword extends** ឬ **implements** នៅ **Class Declaration**។

☞ **Class** ដែលឱ្យគេយក **Property** មកប្រើត្រូវបានគេហៅថា (**Super Class , Base Class** ឬ **Parent Class**)។

☞ □ **Class** ដែលទទួល **Property** ពីគេត្រូវបានគេហៅថា (**Sub Class, Derived Class** ឬ **Child Class**)។

១.១ . សេចក្តីផ្តើម

- ☞ □ **Keyword Extends** ប្រើសម្រាប់ **Inherit** ពី **Class** (**Class Inheritance**)។
- ☞ **Keyword Implements:** ប្រើសម្រាប់ **Inherit** ពី **Interface** (**Interface Inheritance**)។
- **Inheritance:** មានលក្ខណៈ២ប្រភេទគឺ **IS-A Relationship** និង **HAS-A Relationship** ។
- យើងអាច **extend** ពី **Class** បានតែមួយប៉ុណ្ណោះ។
- តែយើងអាច **implement** ពី **Interfaces** មួយ ឬ ច្រើនបាន។

១.២. IS-A Relationship

- **IS-A Relationship:** គឺសំដៅ លើទំនាក់ទំនងរវាង **Sub Class** និង **Super Class** ដែល **Object** នៃ **Sub Class** គឺជាប្រភេទរបស់ **Object** នៃ **Super Class** នោះ ។

```
public class Animal{  
}  
  
public class Mammal extends Animal{  
}  
  
public class Reptile extends Animal{  
}  
  
public class Dog extends Mammal{  
}
```

```
public interface Animal{  
}  
  
public class Mammal implements  
Animal{  
}  
  
public class Reptile implements  
Animal{  
}  
  
public class Dog extends Mammal{  
}
```


១.២. IS-A Relationship

□ ចំពោះ **Object Oriented** យើងអាចបាននិយាយបានថា:

☞ **Animal** គឺជា Super Class របស់ Mammal Class

☞ **Animal** គឺជា Super Class របស់ Reptile Class

☞ **Animal** គឺជា Super Class របស់ Dog Class

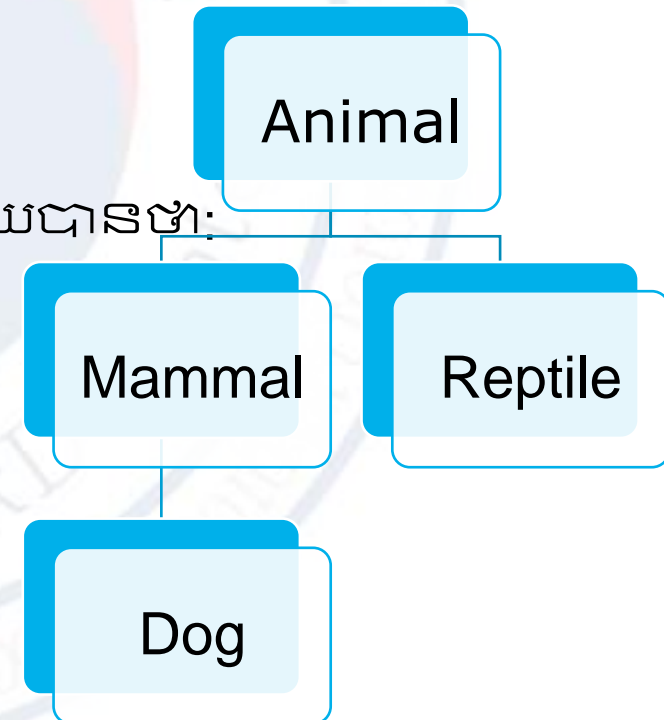
□ ចំពោះ **IS-A relationship** យើងអាចបាននិយាយបានថា:

☞ Mammal **IS-A** Animal

☞ Reptile **IS-A** Animal

☞ Dog **IS-A** Mammal

☞ ដូចនេះ: Dog **IS-A** Animal



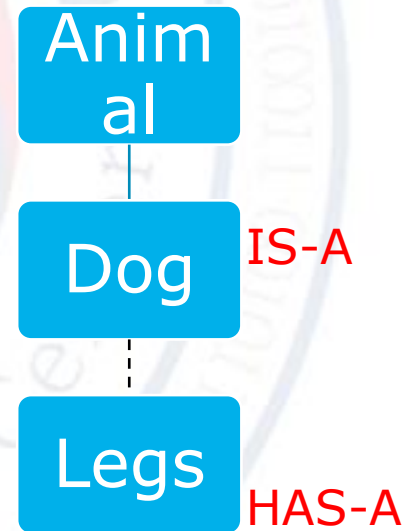
១.៣. HAS-A Relationship

- **HAS-A Relationship:** គឺសំដៅលើទំនាក់ទំនងរវាង **Class** ពីរ ក្នុងការយក **Instance Variable** ដែលយោងទៅ **Class** មួយ មកប្រើក្នុង **Class** មួយទៀត ដោយមិនចាំបាច់សរសេរកូដជ្រុងដែរ ។

```
//Animal.java
public class Animal {}

//Legs.java
public class Legs {}

//Dog.java
public class Dog extends Animal {
    private Legs dogLegs;
}
```



១.៤. The instanceof Keyword

- **Keyword instanceof:** គឺត្រូវបានគេប្រើដើម្បីត្រួតពិនិត្យ **Object** ណាមួយ ថាតើ វាជាប្រភេទនៃ **Class** ឬ **Interface** មួយណា ដោយធ្វើការប្រៀបធៀប **Instance Object** នោះជាមួយនឹង **Class** ឬ **Interface** ដែលផ្តល់ឱ្យហើយវាផ្តល់ លទ្ធផលជា Boolean (true ឬ false)។

លទ្ធផល:

```
true  
true  
true
```

```
interface Animal{}  
  
class Mammal implements Animal{}  
  
public class Dog extends Mammal{  
    public static void main(String args[]){  
  
        Mammal m = new Mammal();  
        Dog d = new Dog();  
  
        System.out.println(m instanceof Animal);  
        System.out.println(d instanceof Mammal);  
        System.out.println(d instanceof Animal);  
    }  
}
```

សមាជិក

ក្រុមអ្នកស្រាវជ្រាវ



ល. ខេង ចាន់វិជ្ជា

vichea@rocketmail.com



ល. សែម ចិត្រា

sabaychitra84@gmail.com



ល. ឡៅ ស៊ុនឡេង

sunlenglao@gmail.com.com



ក. លីម សុខហេង

lim.sokheng1@gmail.



ល. យ៉ូ វណ្ណវិទូ

ravuthz@gmail.com

ក្រុមផលិតវីដេអូ



ល. ឈុន បញ្ញាវត្ត

chhunpanharath@gmail.com



ល. ព្រាប វិទូ

Itpreap.vuthy@gmail.com



ល. ហង្ស បូរី

hounboreyrupp@gmail.com



ល. ហួ ឈុនឡេង

huochhunleng@yahoo.com

មេរៀនបន្ទាប់ នឹងធ្វើការបង្ហាញពី **Overriding**