



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

មេរៀនទី៧: Packages

៧.១. អ្វីទៅជា Package?

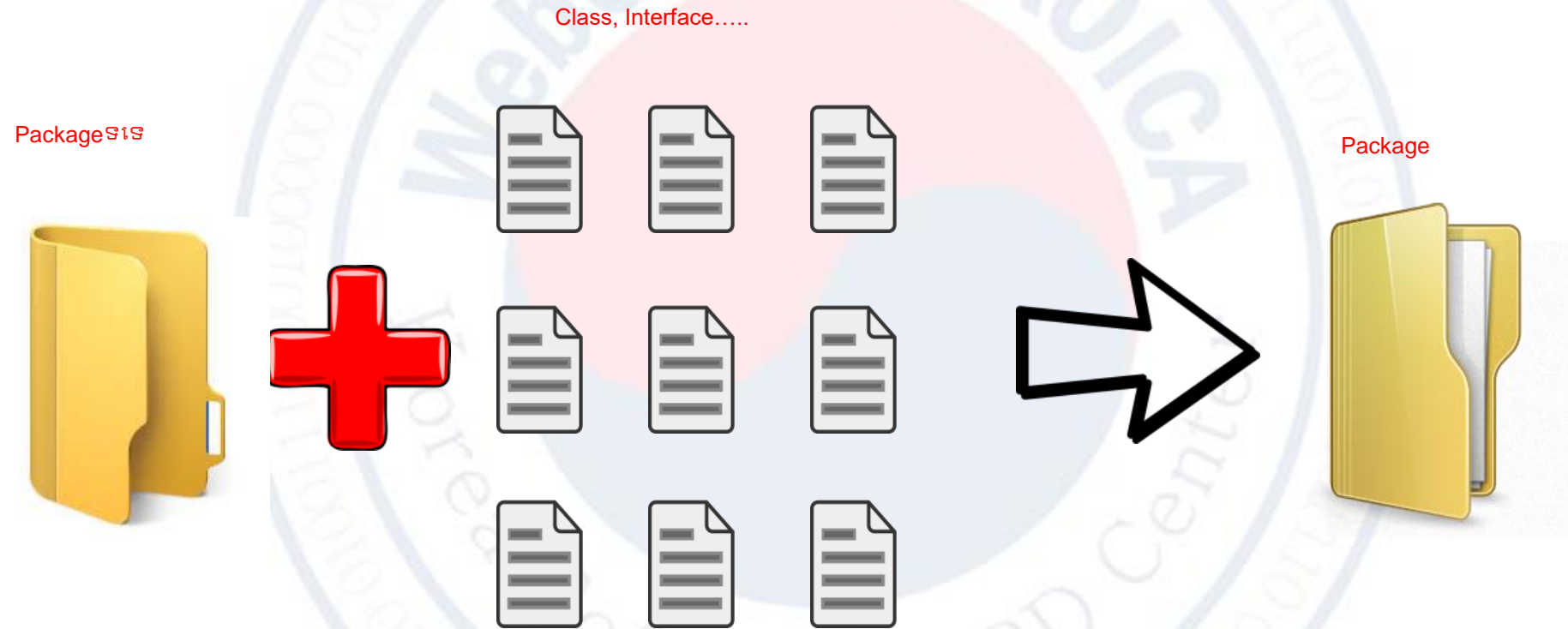
៧.២. ការបង្កើត Package

៧.៣. ការនាំយក Class នៅក្នុង Package មកប្រើ

៧.១. អ្វីទៅជា Package?

- **Package** គឺជាបណ្តុំ នៃ **Classes** ដែលមានទំនាក់ទំនងគ្នា ហើយវាផ្តល់នូវការការពារនៃការចូលប្រើប្រាស់ **Classes** , **Interfaces** និង **Enumerations** នៅក្នុង **package** និងការគ្រប់គ្រងលើ **Name Space** ។ ហើយ **Class** ដែលដាក់នៅក្នុង **Package** មួយ អាចយកប្រើបានតាមរយៈឈ្មោះ **Package** របស់វា។ នៅក្នុង **Package** មួយមិនអាចមាន **Class** ពីរមានឈ្មោះដូចគ្នាឡើយ ហើយគ្រប់ **Class** ទាំងអស់នៅក្នុង **Java** ជារបស់ **Package** ។ កាលណា **Class** ដែលគ្មានឈ្មោះ **Package** នៅពីលើបញ្ជាក់នោះទេ **Default Package** ត្រូវបានប្រើ ។ ហើយ **Default Package** គ្មានឈ្មោះឡើយដែលធ្វើអោយ **Default Package** មើលឃើញ

୩.୧. ୱାଜାଜା Package?



៧.២. ការបង្កើត Package

- ដើម្បីបង្កើត **Package** មួយគេប្រើពាក្យ **Package** នៅផ្នែកខាងលើគេបង្អស់របស់ **file java** ។

ទំរង់ទូទៅនៃការប្រើប្រាស់ **Package** ៖

```
package packagename;
```

៧.២. ការបង្កើត Package

- គេក៏អាចបង្កើត **Package** នៅក្នុង **Package** បានដែរ។
- ដើម្បីធ្វើបែបនេះ គេត្រូវញែកដាច់ពីគ្នានូវឈ្មោះ **Package** មួយពី **Package** មួយទៀតដោយប្រើសញ្ញា (.) ។
- ទំរង់ទូទៅនៃការប្រើប្រាស់ **Package** ច្រើនថ្នាក់គឺ៖

```
package pkg1.pkg2;
```

```
package pkg1.pkg2.pkg3. ... pkgN;
```

៧.២. ការបង្កើត Package

ឧទាហរណ៍ ១

ការដាក់ **Interface** មួយនៅក្នុង **Package** ឈ្មោះ: **Animals**

```
/* File name : Animal.java */  
package animals;  
  
interface Animal {  
    public void eat();  
    public void travel();  
}
```

៧.២. ការបង្កើត Package

ឧទាហរណ៍ ២

ការបង្កើត **Class** ឈ្មោះ: **Test** ដាក់ក្នុង Package ឈ្មោះ: **Mine**

```
package Mine;  
  
public class Test{  
    public void display(){  
        System.out.println("Welcome to World of Java");  
    }  
}
```


៧.៣. ការនាំយក Class នៅក្នុង Package មកប្រើ

- ទំរង់ទូទៅនៃឃ្លា **Import** គឺ៖ `import packagename.classname;`
- ក្នុងនោះ **Packagename** ជាឈ្មោះរបស់ **Package** ដែលអាចបញ្ជូនរួមទាំង **Path** របស់វាដោយមានការបញ្ជាក់យ៉ាងពេញលេញ ហើយនិង **ClassName** គឺជាឈ្មោះរបស់ **Class** ដែលត្រូវនាំយកមកប្រើ ។ បើសិនជាយើងចង់នាំយក **Class** ទាំងអស់ដែលមាននៅក្នុង **Package** យើងត្រូវប្រើសញ្ញា (*) ជំនួសអោយឈ្មោះ `class`។

ឧទាហរណ៍

```
import pkg1.Myclass;
```

```
ឬ import pkg1. * ;
```

៧.៣. ការនាំយក Class នៅក្នុង Package មកប្រើ

- ការ Import Package មកប្រើក្នុង Class ឈ្មោះ Packs

```
import Mine.*;           //Package ត្រូវបានហៅមកប្រើ

class Packs{
    public static void main(String args[]){
        Test d=new Test();

        // Object របស់ class Test ត្រូវបាន
        d.display();
    }
}
```

បញ្ចប់

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

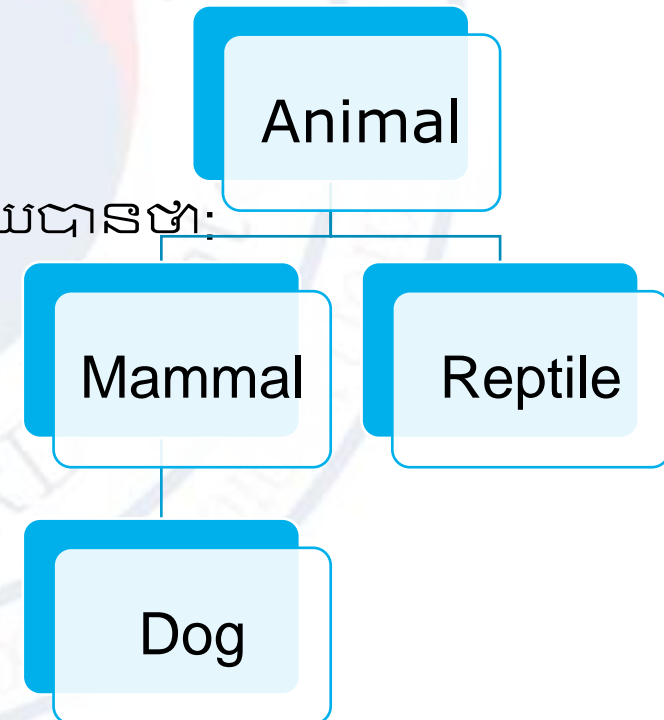
□ ចំពោះ **IS-A relationship** យើងអាចបាននិយាយបានថា:

☞ Mammal **IS-A** Animal

☞ Reptile **IS-A** Animal

☞ Dog **IS-A** Mammal

☞ ដូចនេះ: Dog **IS-A** Animal



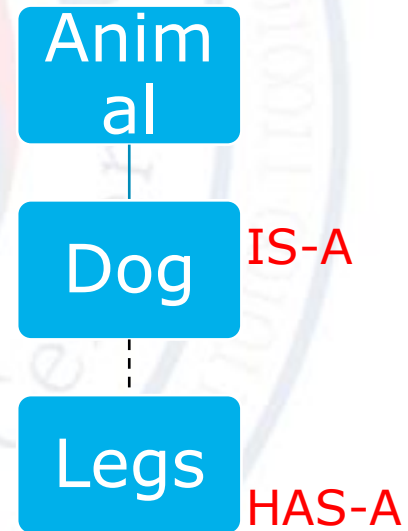
១.៣. HAS-A Relationship

- **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. គ្រី សុផាន់ណាត់
krysophanatt@gmail.com



ល. ង៉ោ ហ្គេចឡេង
ngorgechleng@gmail.com



ល. ប្រាក់ ដារ
prakda99@yahoo.com



ក. សុង ដារតនា
songdarathana@gmail.com



ល. អេ កុសល
longkosal7@gmail.com

ក្រុមផលិតវីដេអូ



ល. ឈុន បញ្ហាភត្ត
chhunpanharath@gmail.com



ល. ហង្ស បូរី
houngboreyrupp@gmail.com



ល. ត្រាប វិថ្នី
Itpreap.vuthy@gmail.com



ល. ហួ ឈុនឡេង
huochhunleng@yahoo.com

មេរៀនបន្ទាប់ នឹងធ្វើការបង្ហាញពី Overriding