

Part



8

Java Programming Language
Mr.Rungrote Phonkam
rungrote@it.kmitl.ac.th



Contents

1. Package
2. Access Level
3. Inheritance
4. super vs this
5. Polymorphism



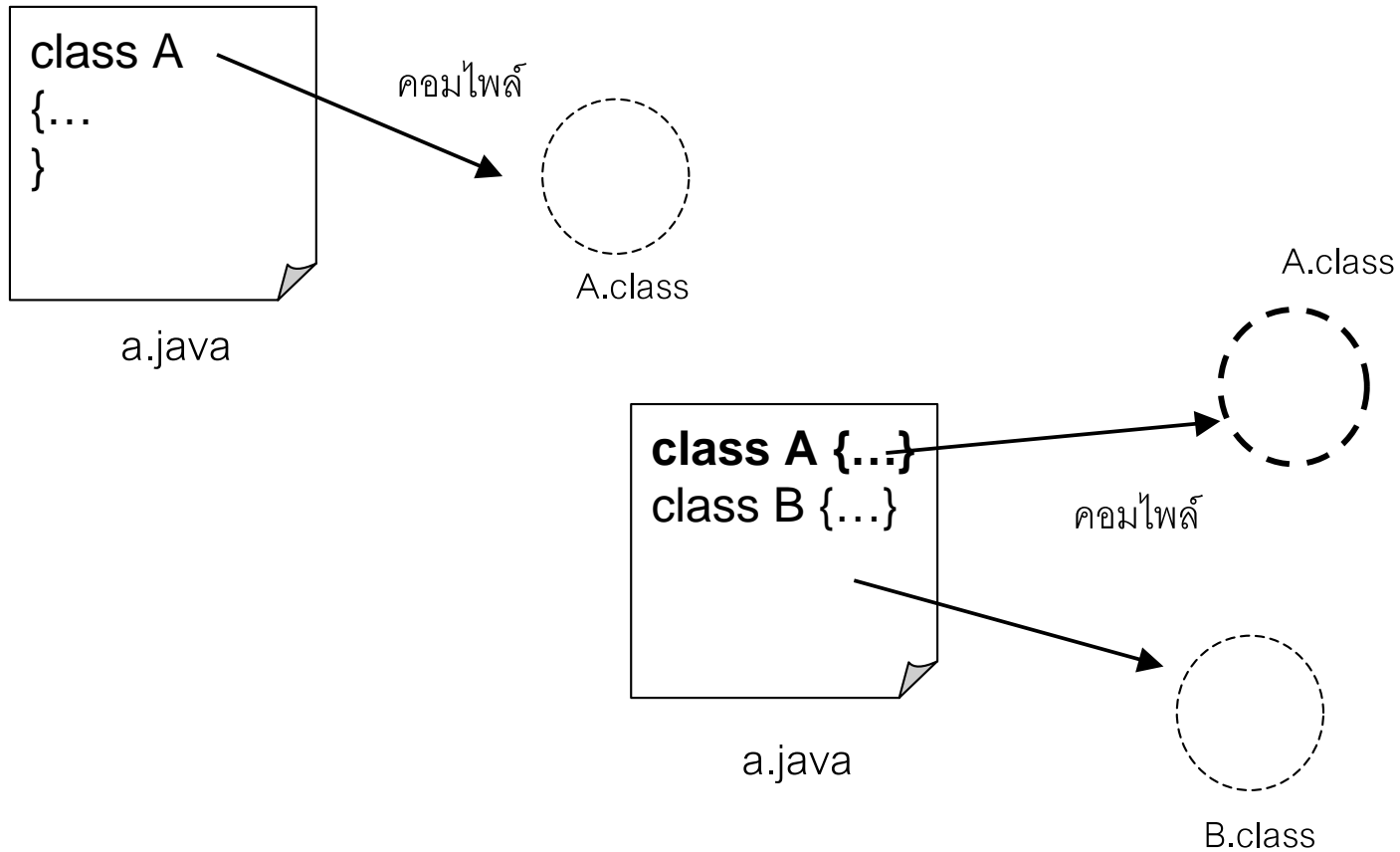
1. Package

แพ็คเกจ (Package)

- คือการรวมกลุ่มของคลาสไว้ในชุดเดียวกัน
- เพื่อความสะดวกในการอ้างอิงเพื่อใช้งาน
- ทำให้การจัดเก็บ Bytecodes เป็นไปอย่างมีระบบ
- ใช้หลักเกณฑ์การจัดแพ็คเกจ เป็นไปตามการจัดไดเรกทอรีของระบบ
- ทำให้เกิดรูปแบบหนึ่งของการกำหนดระดับการเข้าถึง



1. Packages





1. Package

การสร้างแพ็คเกจ

รูปแบบ

```
package Package_Name
```

เมื่อ

package

คือเวิร์ดสำหรับระบุความเป็นแพ็คเกจของซอร์สโค้ด

Package_Name

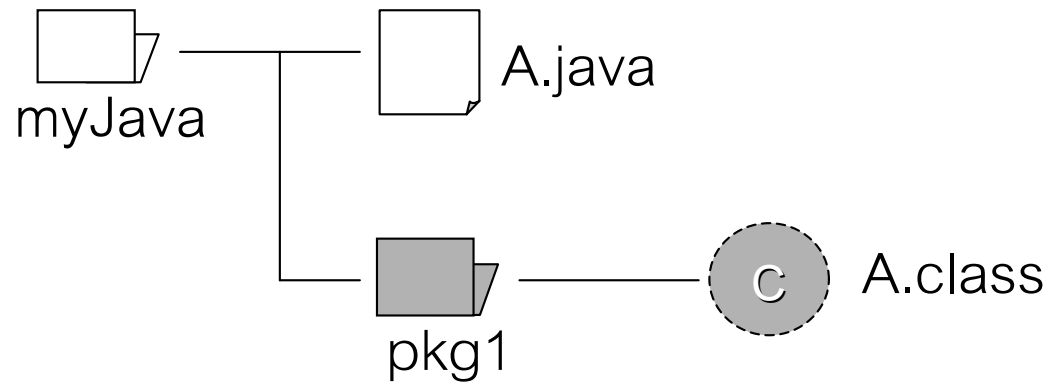
คือชื่อแพ็คเกจ

```
package Pkg1;  
public class A  
{ ...  
}
```



1. Package

```
c:\myJava>javac -d . a.java
```





1. Package

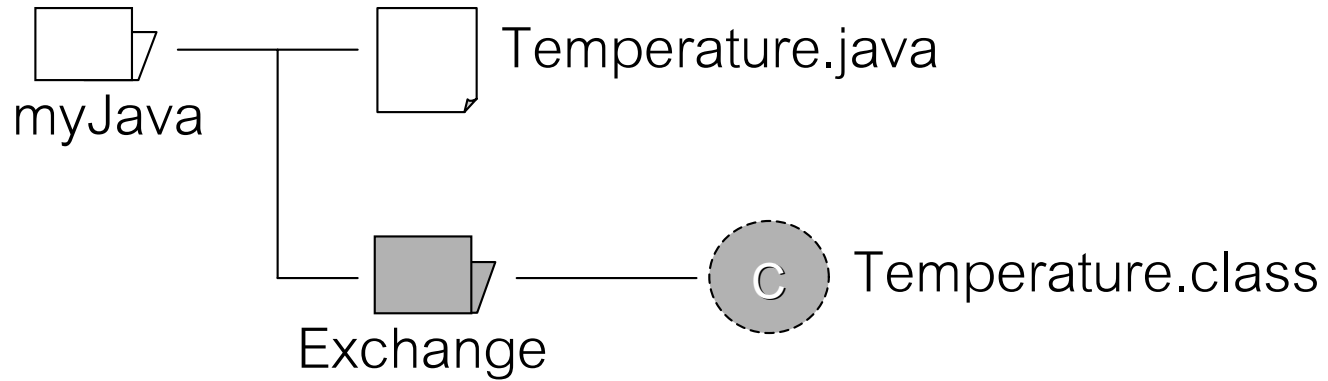
```
package Exchange;
```

```
public class Temperature
```

```
{  
    public static float F2C(float f)  
    {  
        return ((f-32) * 5.0F) / 9.0F;    }  
    public static float C2F(float c)  
    {  
        return (c * 9.0F/5.0F) + 32.0F;    }  
}
```



1. Package





1. Package

การใช้งานแพ็คเกจ

รูปแบบ

```
import Package_Name.Class_Name
```

เมื่อ

import คือคีย์เวิร์ดสำหรับการเรียกใช้งานคลาสจากแพ็คเกจอื่น
Package_Name คือชื่อแพ็คเกจรวมถึงเส้นทาง(กรณีมีหลายแพ็คเกจย่อย)
Class_Name คือชื่อคลาส

หมายเหตุ การหาคลาสเริ่มจากใดเรียกทอริปัจจุบัน
และที่ระบบใน CLASSPATH

```
import Pkg1.A;  
public class A  
{ ...  
}
```



1. Package

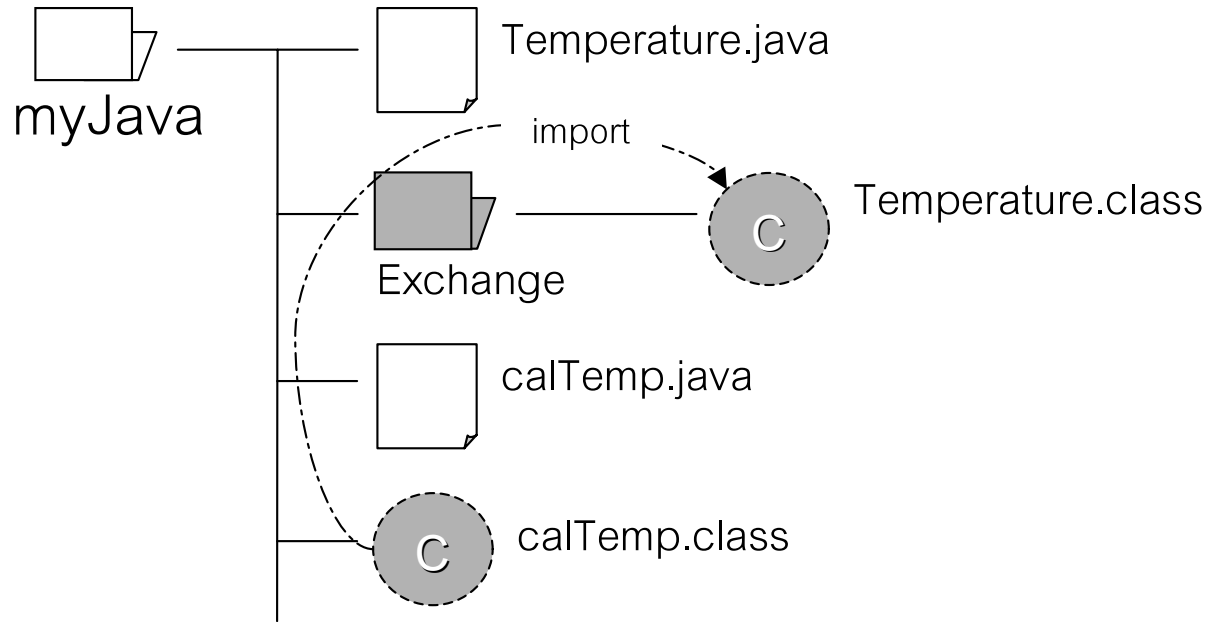
```
import Exchange.Temperature;
class CalTemp
{   public static void main(String args[])
    {       float t = Float.parseFloat(args[0]);
            System.out.print(args[0] + " Degree Centigrade = ");
            System.out.print(Temperature.C2F(t));
            System.out.println(" Degree Fahrenheit");
    }
}
```

```
java CalTemp 32.0
```

```
32.0 Degree Centigrade = 89.6 Degree Fahrenheit
```



1. Package





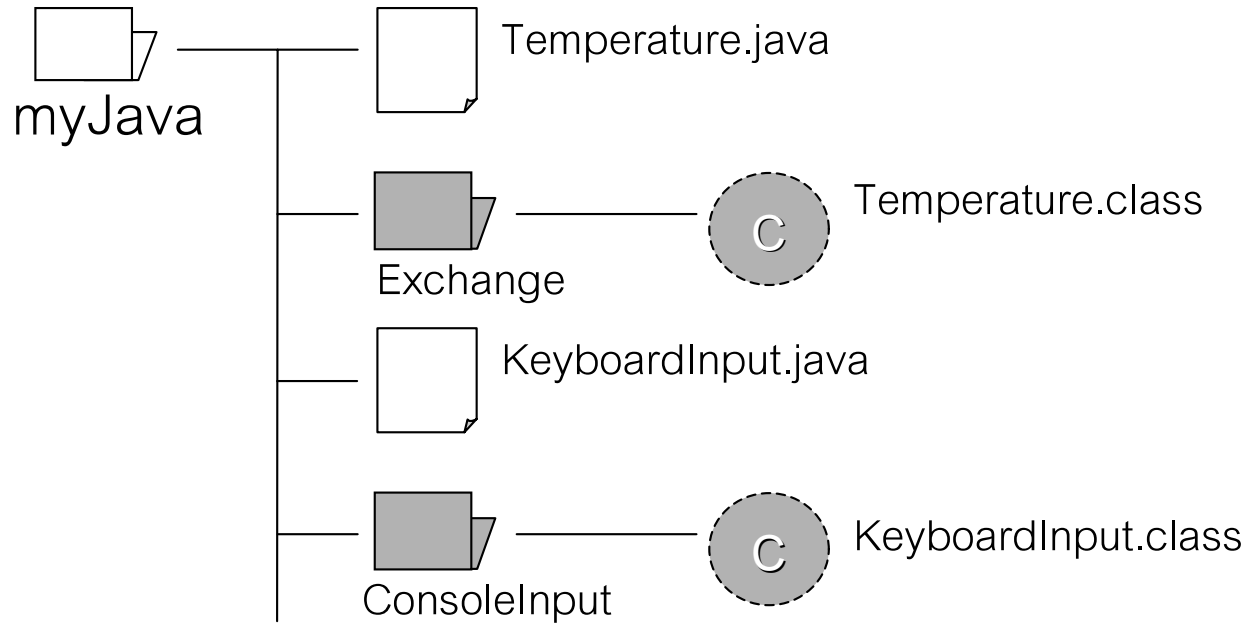
1. Package

```
package ConsoleInput;
import java.io.*;

public class KeyboardInput
{   public static String readText()
    {   String name = "";
        BufferedReader inReader =
        new BufferedReader(new InputStreamReader(System.in));
        try {   name = inReader.readLine();   }
        catch(Exception e) {   e.printStackTrace();   }
        finally {   return name;   }
    }
}
```



1. Package





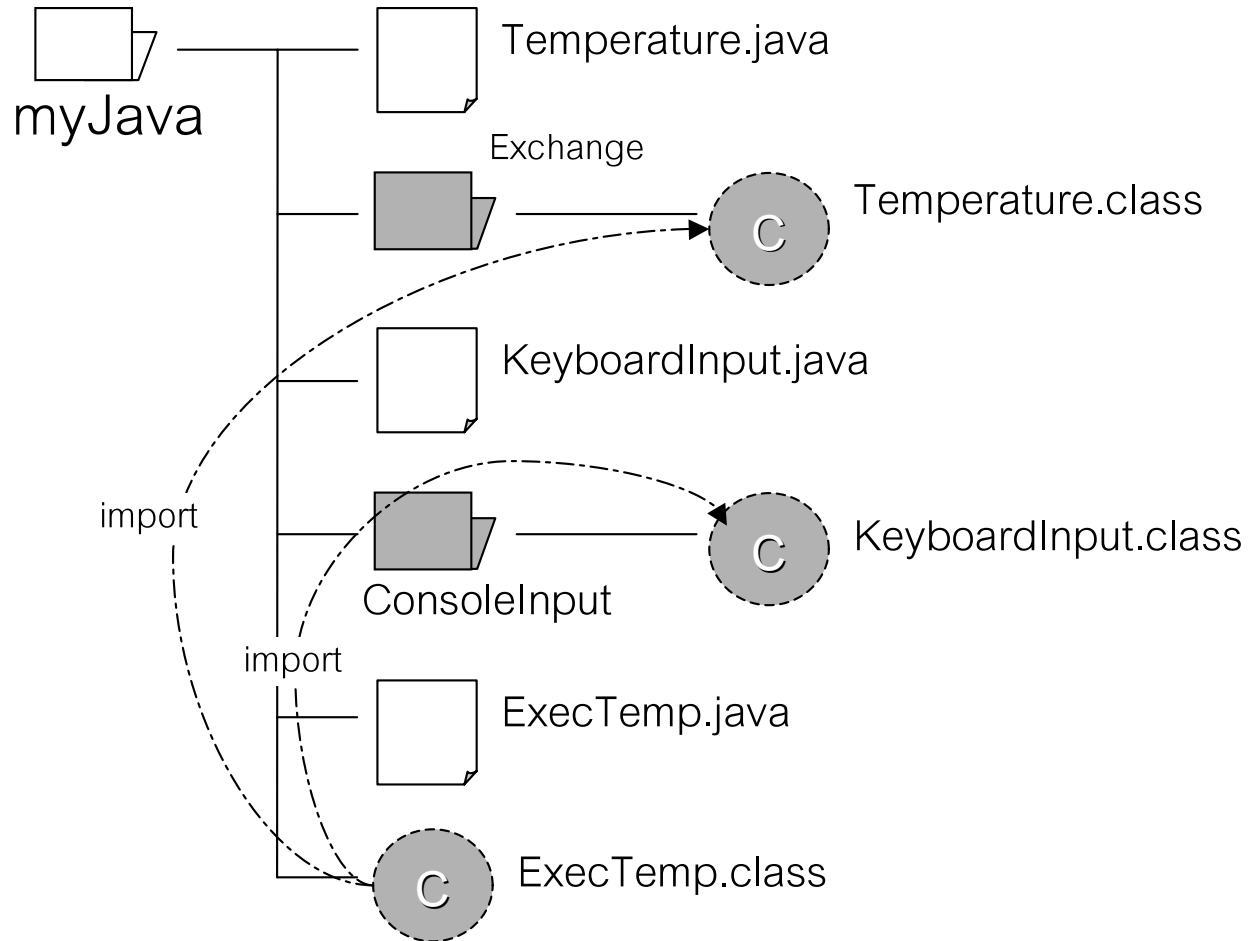
1. Package

```
import ConsoleInput.KeyboardInput;
import Exchange.Temperature;
class ExecTemp
{   public static void main(String args[])
    {   int t;
        System.out.print("Enter Degree Cencert Temperature: ");
        t = Integer.parseInt(KeyboardInput.readText());
        System.out.println(t + " Degree Conceit = " +
            Temperature.C2F(t) + " Farenheigh");
    }
}
```

```
java ExecTemp
Enter Degree Cencert Temperature: 100
100 Degree Conceit = 212.0 Farenheigh
```



1. Package





2. Access Level

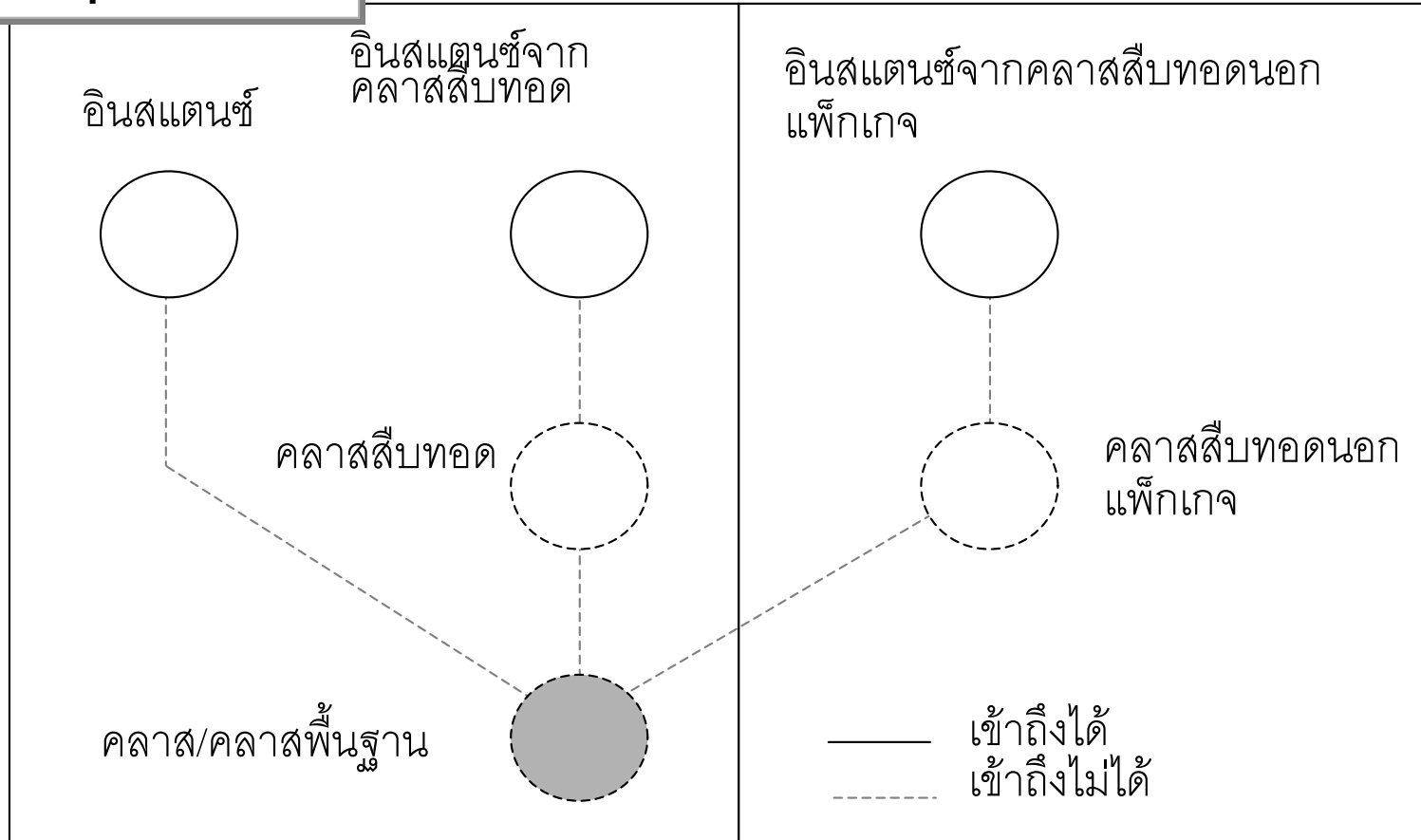
ระดับการเข้าถึง (Accessibility/ Access Level)

- เป็นการนำเอาแนวคิดเกี่ยวกับ Information Hiding มาใช้งาน
- แบ่งการเข้าถึงเป็น 4 ระดับคือ **private**, **package**, **protected** และ **public**
- สามารถกำหนดให้กับทั้ง Data Member และ Method Member ภายในคลาส หรือระบุให้กับคลาสโดยตรง
- การระบุระดับการเข้าถึงถือเป็นระบบความปลอดภัยลักษณะหนึ่งที่ใช้กับคลาส



2. Access Level

private



ระดับการเข้าถึงแบบ private



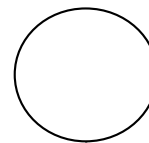
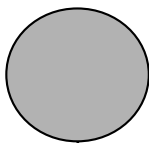
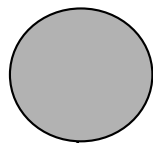
2. Access Level

package

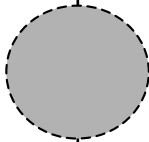
อินสแตนซ์

อินสแตนซ์
จากคลาสสืบทอด

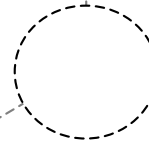
อินสแตนซ์จากคลาสสืบทอดนอก
แพ็คเกจ



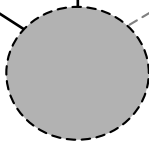
คลาสสืบทอด



คลาสสืบทอดนอก
แพ็คเกจ



คลาส/คลาสพื้นฐาน



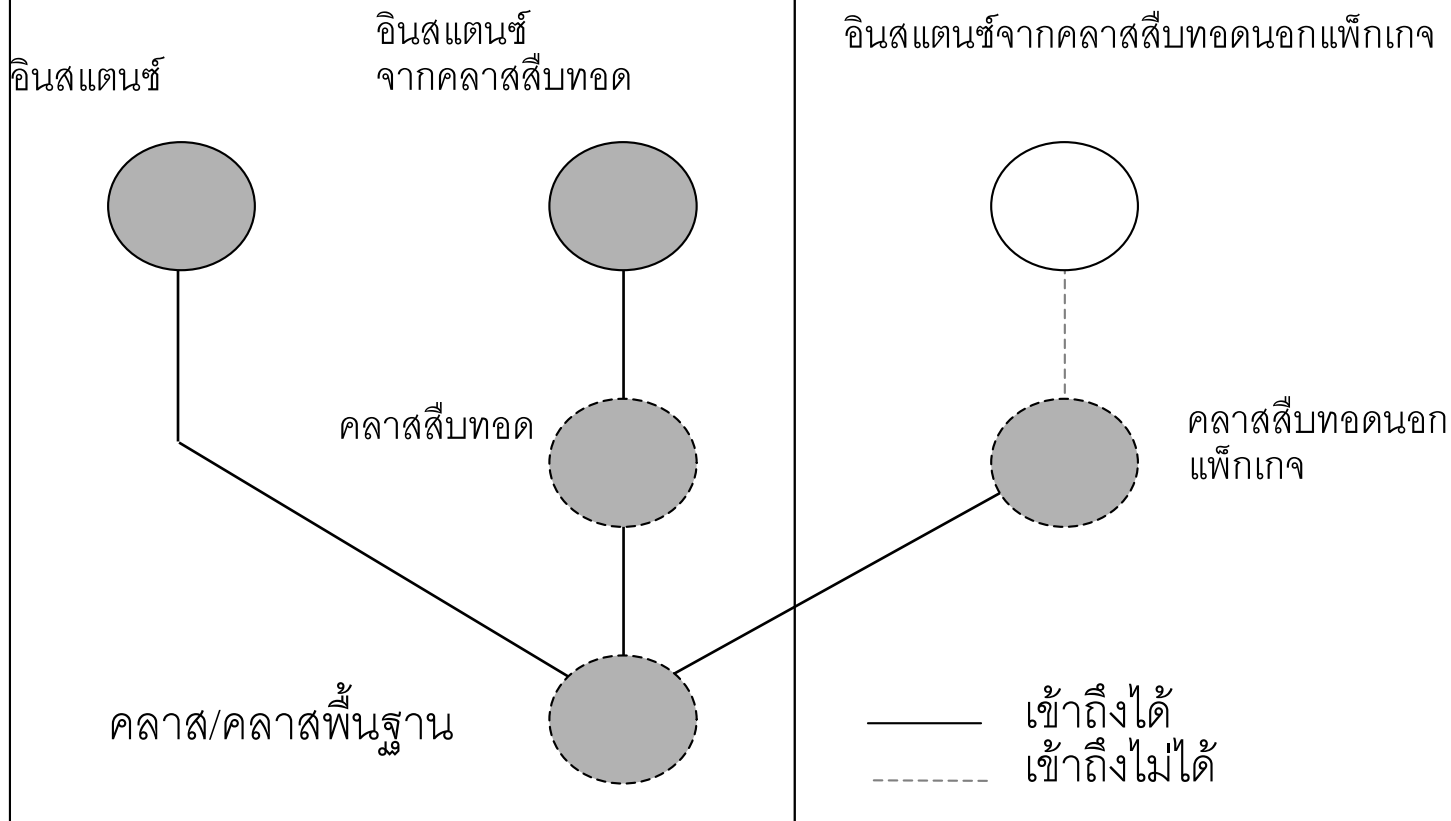
— เข้าถึงได้
- - - เข้าถึงไม่ได้

ระดับการเข้าถึงแบบ package(default)



2. Access Level

protected

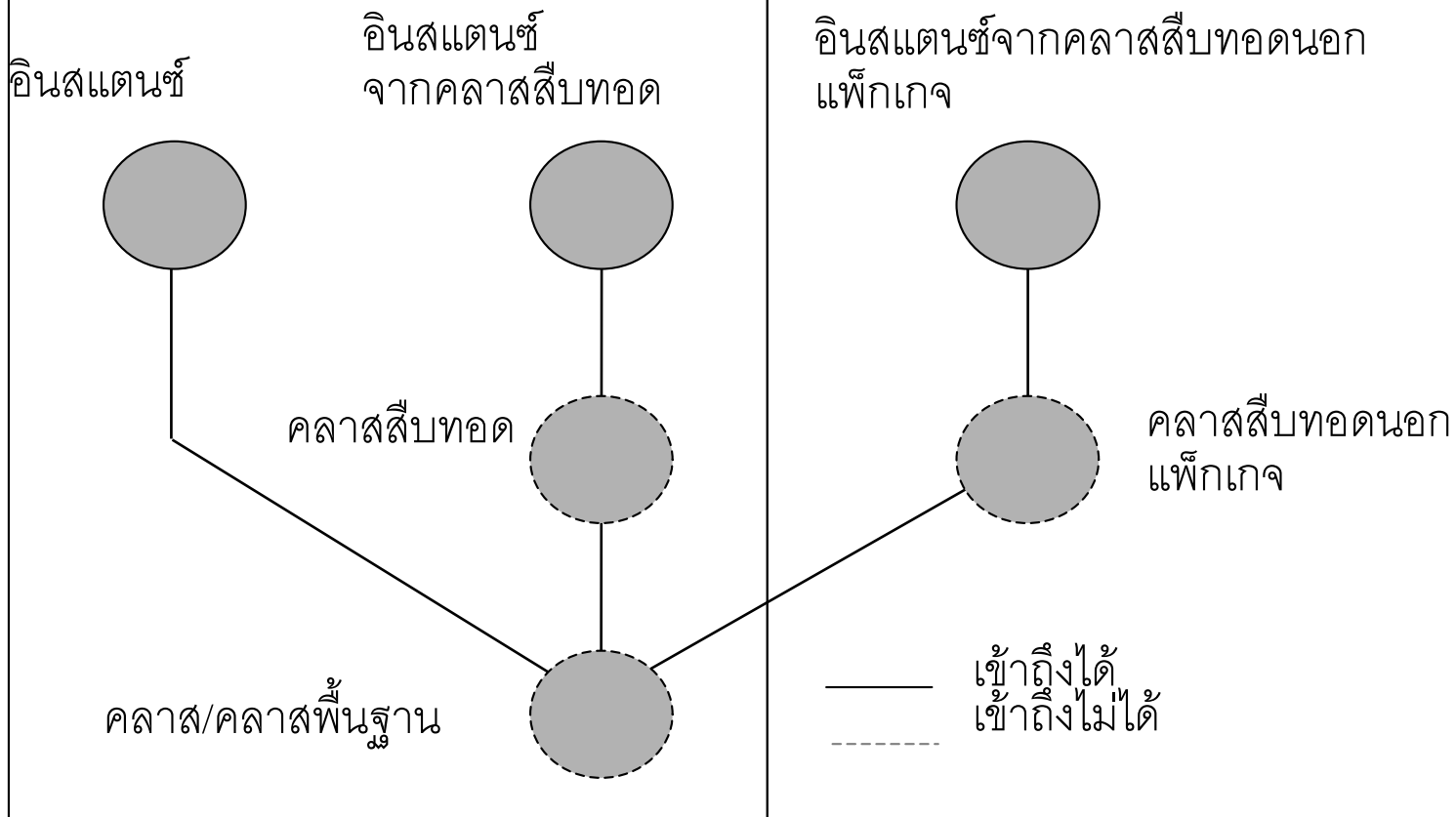


ระดับการเข้าถึงแบบ protected



2. Access Level

public



ระดับการเข้าถึงแบบ public



3. Inheritance

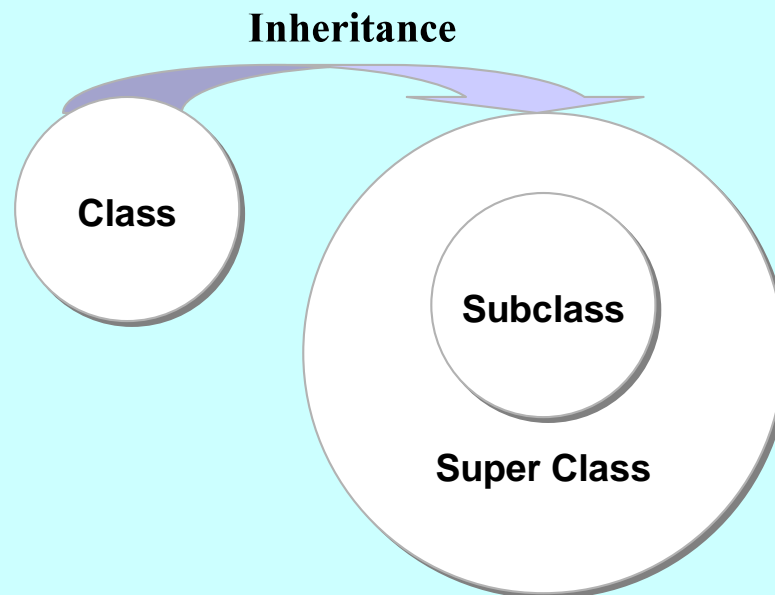
การสืบทอด

- คือการสร้างคลาสใหม่ที่ดึงเอา Data และ Method จากคลาสใดๆมาใช้งาน โดยมีการแทรกหรือปรับปรุง Data และ Method ใหม่เข้าไปในคลาส
- คลาสใหม่หรือคลาสสืบทอดในที่นี้เรียกว่า Subclass หรือ Extend Class
- คลาสใดๆในที่นี้เรียกว่า Super Class
- การสืบทอดทำให้หลักการ Reusability ในแนวคิดของ Object Oriented เป็นจริง
- การทำความเข้าใจกับการสืบทอด ต้องทำความเข้าใจกับระดับการเข้าถึงควบคู่กันไปด้วย



3. Inheritance

ลักษณะการสืบทอดที่เกิดขึ้น





3. Inheritance

รูปแบบ

```
class Subclass extends Super_Class  
{  
    Data_Member;  
    Method_Member;  
}
```

เมื่อ

Subclass	คือชื่อคลาสสืบทอด
Super_Class	คือชื่อคลาสที่ต้องการนำมาใช้สร้างคลาสสืบทอด
extends	คือคีย์เวิร์ดที่ใช้บอกว่าเป็นการสร้างชั้นคลาส
<i>Data_Member</i>	คือดาต้าของชั้นคลาส
<i>Method_Member</i>	คือเมธอดของชั้นคลาส



3. Inheritance

Super Class

```
class Sheet
{
    private int page = 0;
    private String article;
    public String size;
    Sheet()          {      article = ""; size = "A4";      }
    Sheet(String a)  {      article = a;          }
    Sheet(String a, int p)
    { article = a; page = p;      }
    protected void addPage(int p)
    { page += p;      }
    protected int getPage()
    { return page;      }
    protected void printSummary()
    { System.out.println( article + " Contains " + page + " Pages") }
}
```




3. Inheritance

```
class Magazine1 extends Sheet
{
    String name = "";      int issue, year;
    Magazine1(String t, int is, int y)
    {
        name = t; issue = is; year = y;      }
    void printMagazine()
    {
        System.out.println("Magazine Name: " + name);
        System.out.println("Issue: " + issue + " Year: " + year);
        System.out.println("\tPage: " + getPage( ) );
    }
}

class KidMag
{
    public static void main(String arg[])
    {
        Magazine1 mag = new Magazine1("Child Worlds", 5, 2000);
        mag.addPage(20); mag.printMagazine();
    }
}
```



3. Inheritance

java KidMag
Magazine Name: Child Worlds
Issue: 5 Year: 2000
Page: 20



3. Inheritance

```
class Magazine2 extends Sheet
{
    String name = "";
    public String size = "";    // Duplicate Name in Super Class
    int issue, year;
    Magazine2(String t, int is, int y)
    {
        name = t; issue = is; year = y; }
    protected void printSummery()    // Duplicate Name in Super Class
    {
        System.out.println( "Magazine Name: " + name);
        System.out.println( "Issue: " + issue + " Year: " + year);
        System.out.println( "Size:" + size);
    }
}

class AdultMag
{
    public static void main(String arg[])
    {
        Magazine2 mag = new Magazine2("Car Racing", 16, 2020);
        mag.addPage(20); mag.size = "Short A4";
        mag.printSummery();
    }
}
```



3. Inheritance

java AdultMag
Magazine Name: Car Racing
Issue: 16 Year: 2020
Size: Short A4



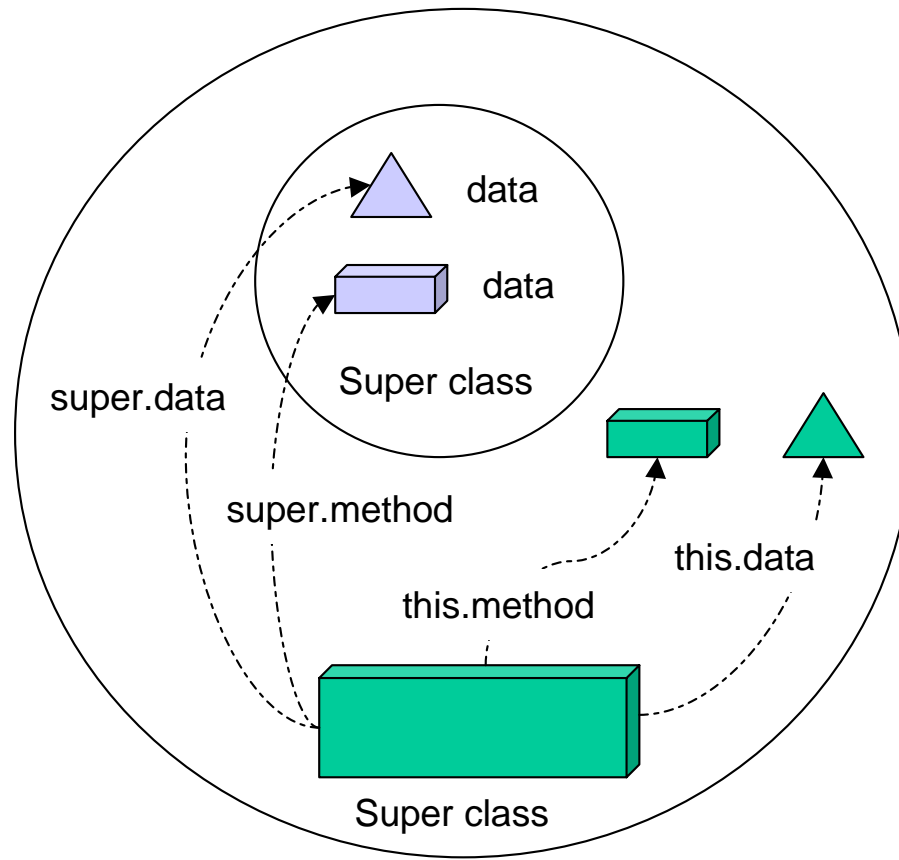
4. super vs this

super vs this

- เนื่องจากการสับสน ในบางครั้งชื่อ data หรือ method มีชื่อเหมือนกัน ทั้งของ super class และ sub class
- ในการอ้างถึง data หรือ method ของ super class สามารถใช้คีย์เวิร์ด super อ้างถึงได้ภายใน sub class
- ในการอ้างถึง data หรือ method ของ sub class สามารถใช้คีย์เวิร์ด this เพื่อชี้เฉพาะว่าเป็น data หรือ method ของ sub class



4. super vs this





4. super vs this

```
class Magazine3 extends Sheet
{
    String name = "";
    public String size = "";    // Duplicate Name in Super Class
    int issue, year;
    Magazine2(String t, int is, int y)
    {
        name = t; issue = is; year = y; }
    protected void printSummery()    // Duplicate Name in Super Class
    {
        System.out.println( "Magazine Name: " + name);
        System.out.println( "Issue: " + issue + " Year: " + year);
        System.out.println( "Size:" + super.size);
    }
}

class AdultMag
{
    public static void main(String arg[])
    {
        Magazine2 mag = new Magazine2("Car Racing", 16, 2020);
        mag.addPage(20); mag.size = "Short A4"; mag.printSummery();
    }
}
```



4. super vs this

java AdultMag
Magazine Name: Car Racing
Issue: 16 Year: 2020
Size: A4



4. super vs this

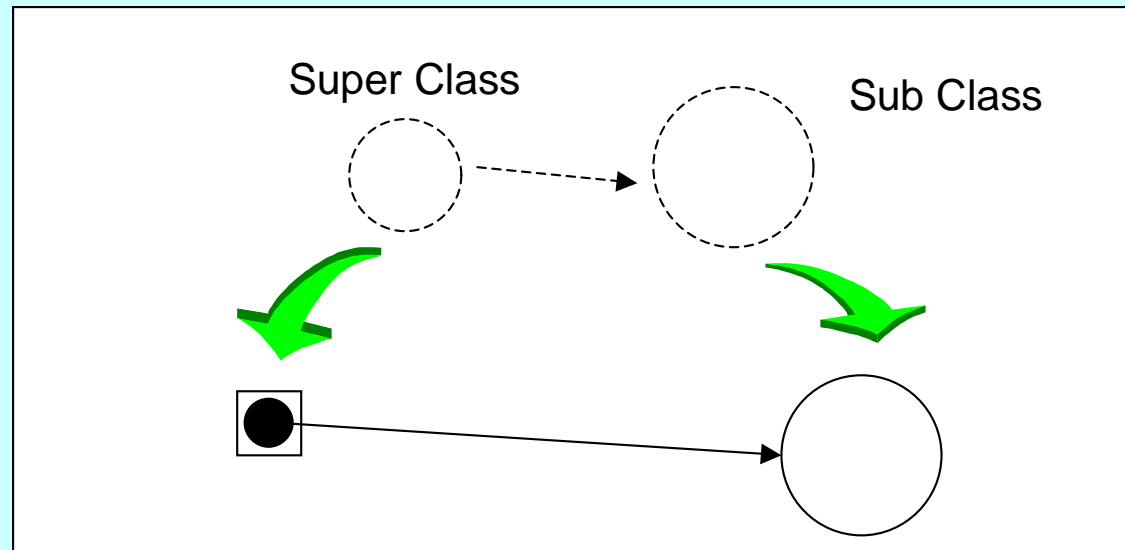
```
class Magazine3 extends Sheet
{
    ....
    protected void printSummery()
    {
        System.out.println( "Magazine Name: " + name);
        System.out.println( "Issue: " + issue + " Year: " + year);
        System.out.println( "Size:" + this.size);
    }
    .....
    .....
}
```



5. Polymorphism

โพลีมอร์ฟี่

- การกำหนดตัวแปรอ้างอิงถึงจาก Super Class และอ้างอิงไปยังอินสแตนซ์ของ Sub Class
- ลักษณะที่สามารถอ้างอิงถึง Data หรือ Method ผ่านทางตัวแปรอ้างอิงถึง





5. Polymorphism

```
class WomanMag
{
    static void printOfMag1(Sheet m)
    {
        m.printSummery();
    }
    static void printOfMag2(Sheet m)
    {
        m.printSummery();
    }
    public static void main(String arg[])
    {
        Sheet obj1 = new Magazine1("Thai Food Cooking", ↪
            2, 2543);
        Sheet obj2 = new Magazine2("How to Shopping", ↪
            2120, 2545);
        printOfMag1(obj1);
        System.out.println("-----");
        printOfMag2(obj2);
    }
}
```



5. Polymorphism

java WomanMag
Contains 0 Pages

Magazine Name: How to Shopping
Issue: 2120 Year: 2545
Size:



5. Polymorphism

