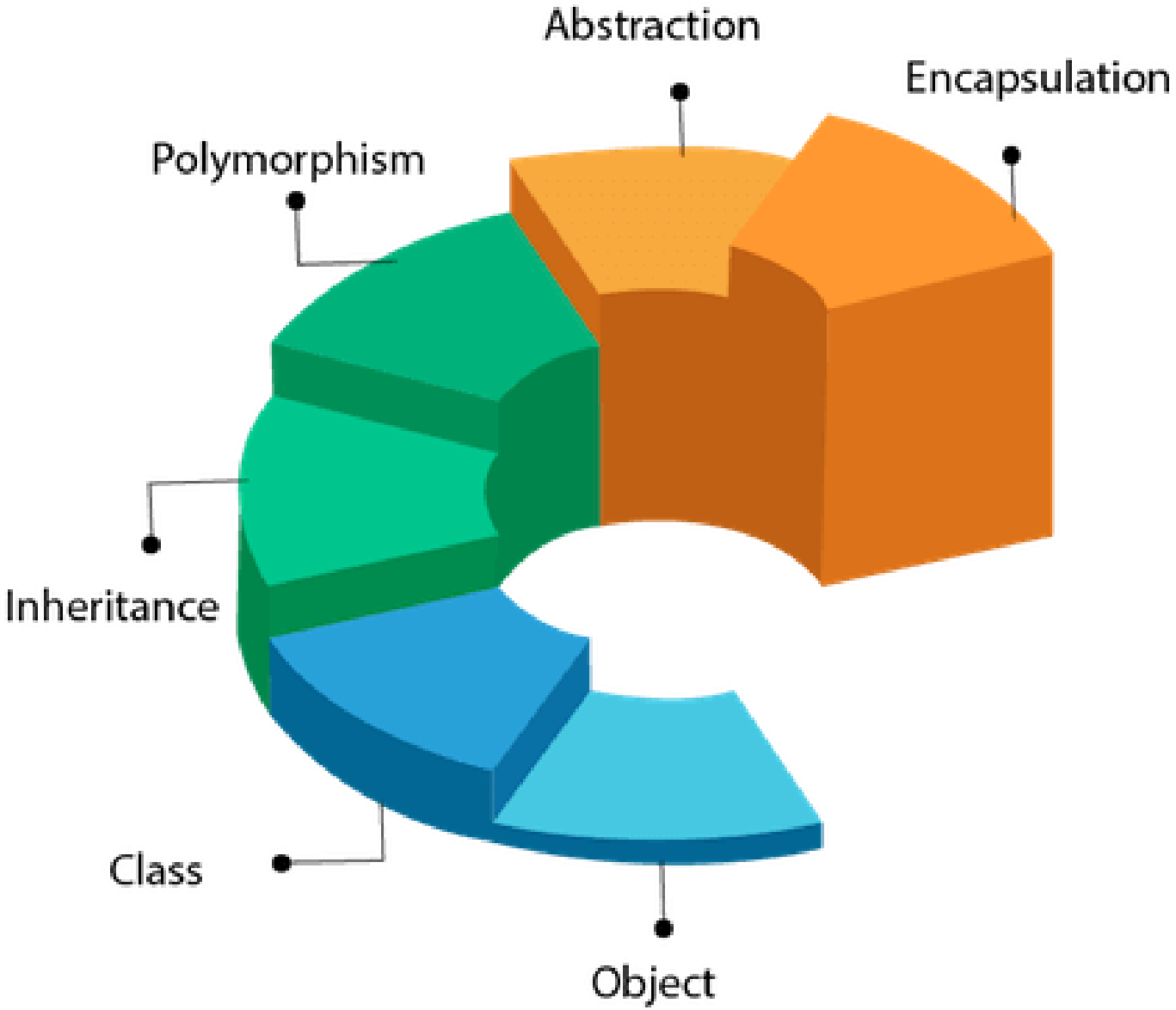
Java OOP

P. SuLAiMAn

OOPs (Object-Oriented Programming System)

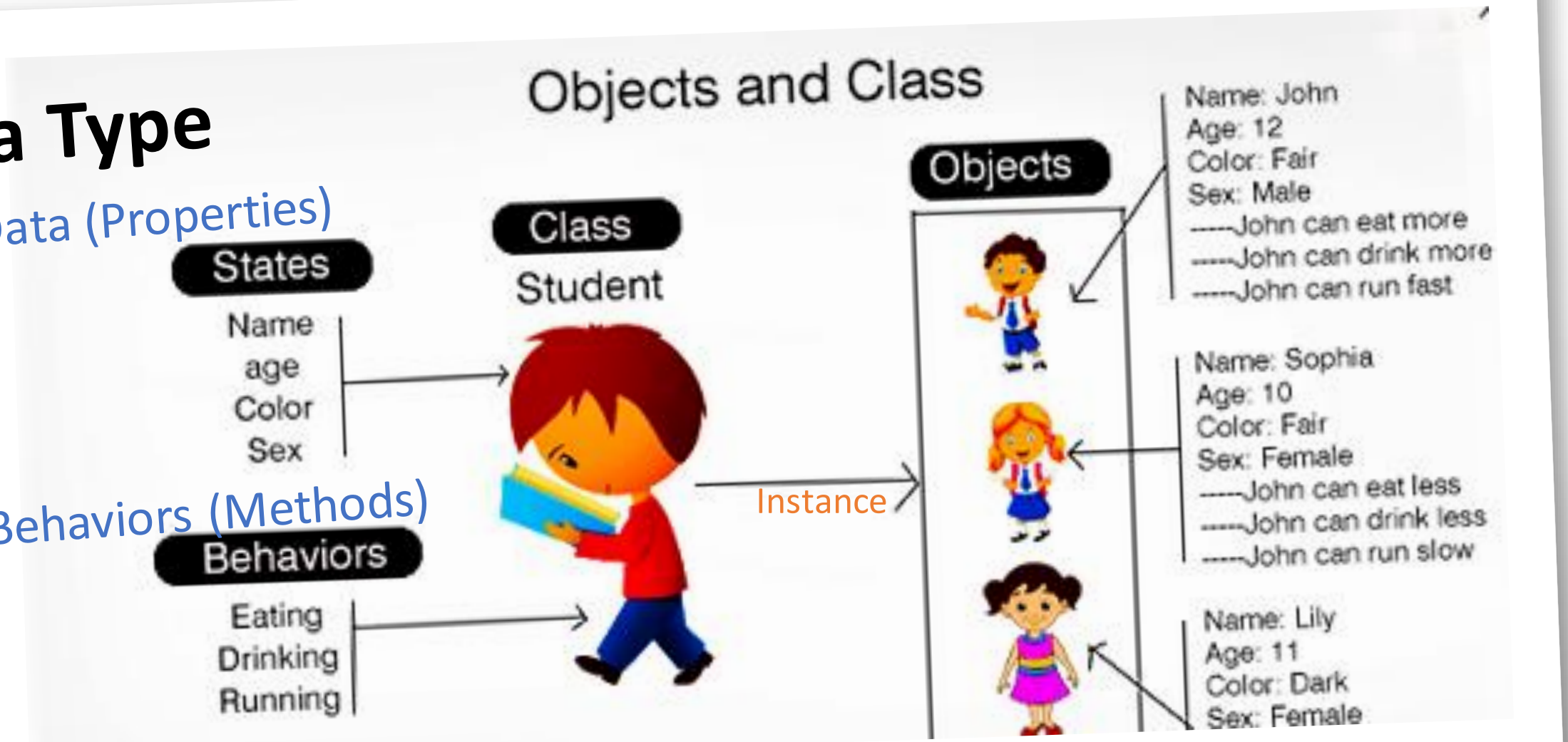


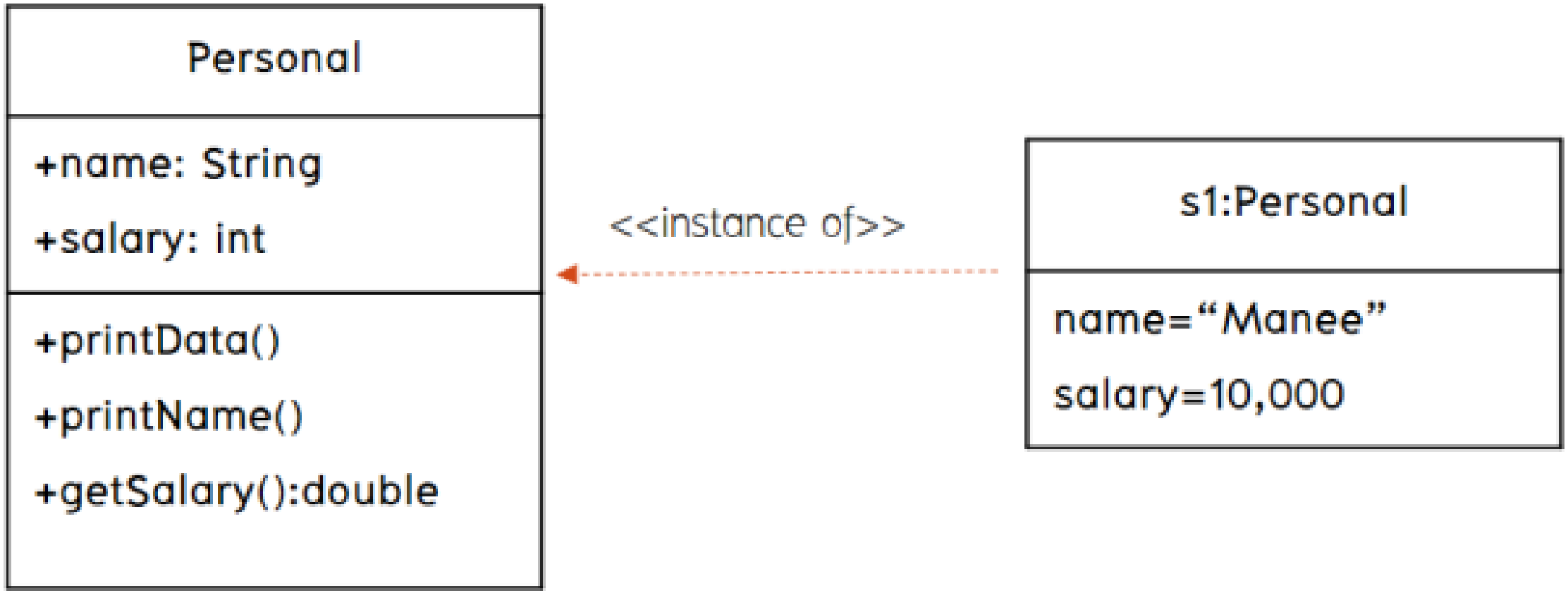

1. OOP Can Support Large Software Project.
2. OOP Offers Better Data Protection.
3. Code Re-Usability.
4. Better Representation Of Real world Objects.
5. Better Software Maintenance.
6. Enhanced Security.
7. Easy Code Modification.
Class
Object
OOP

Data Type

Data (Properties)

Behaviors (Methods)




Personal
+name: String
+salary: int
+printData()
+printName()
+getSalary():double
<<instance of>>
s1:Personal
name="Manee"
salary=10,000

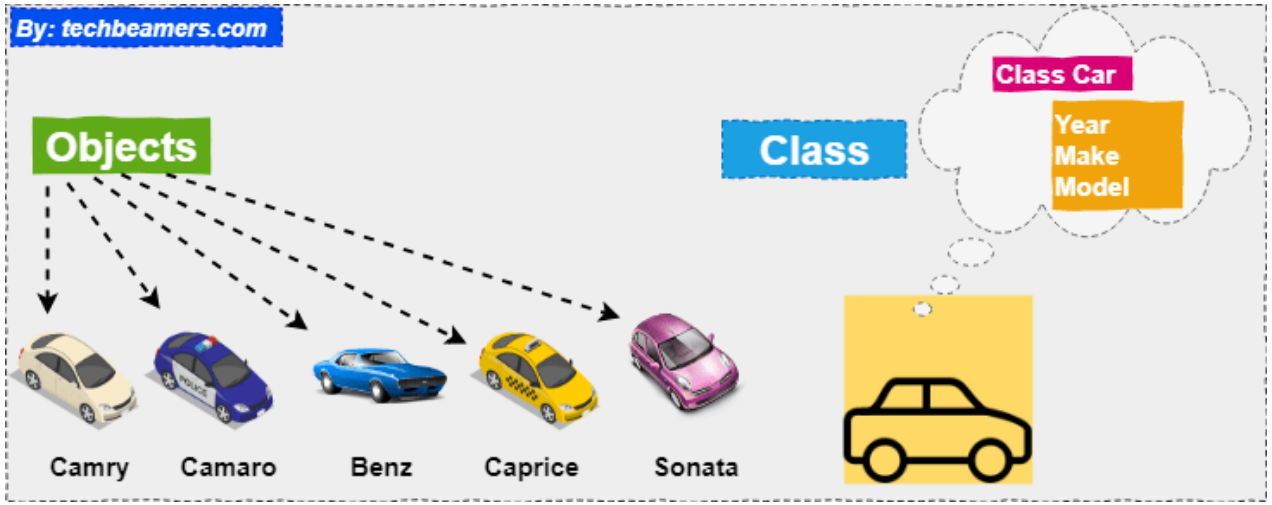
- ทุกอย่างคืออ็อบเจ็คต ซึ่งแต่ละออบเจ็คตมีหน้าที่และความสามารถต่างกัน

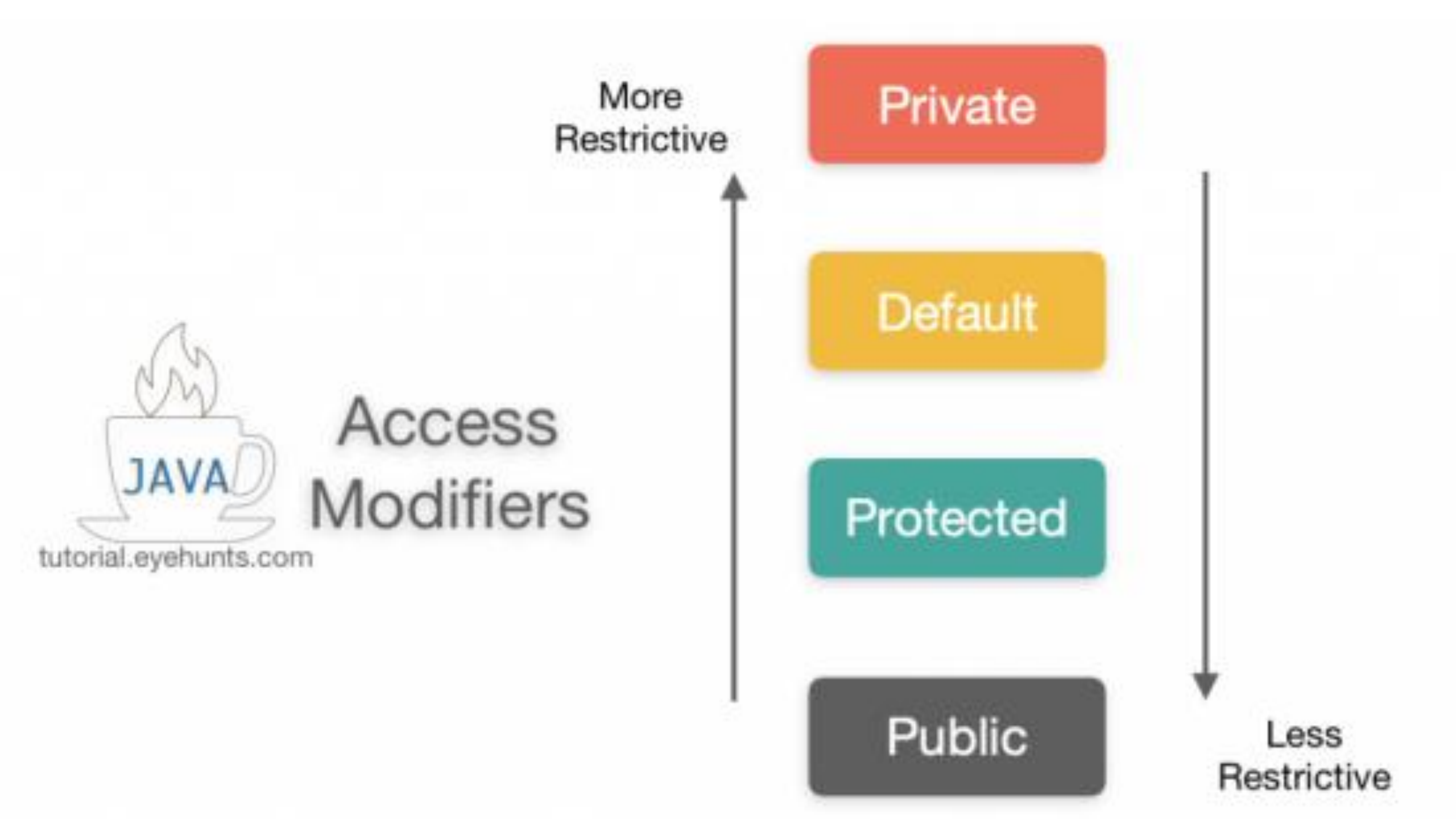
- โปรแกรมเกิดจากการนำออบเจ็คตมาทำงานร่วมกัน

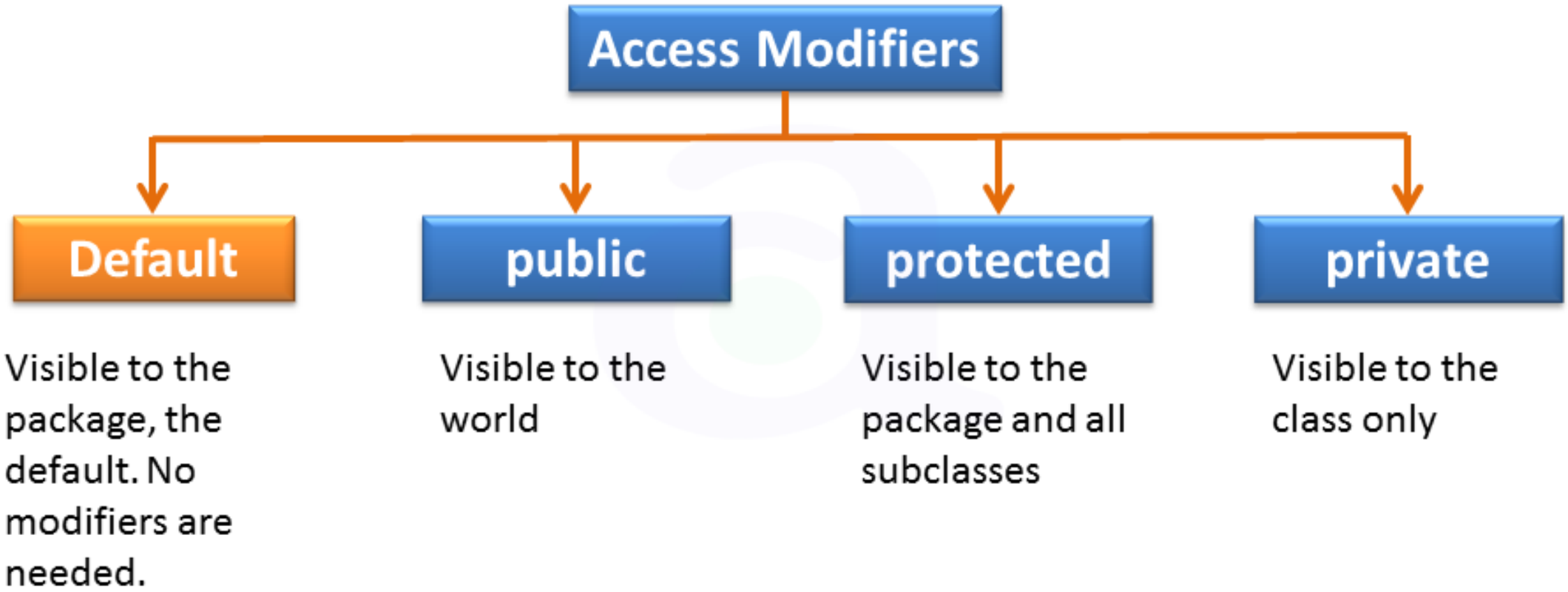
- แต่ละออบเจ็คตเป็นอินสแตนซของคลาส

- แต่ละออบเจ็คตมีสถานะเป็นของตนเอง

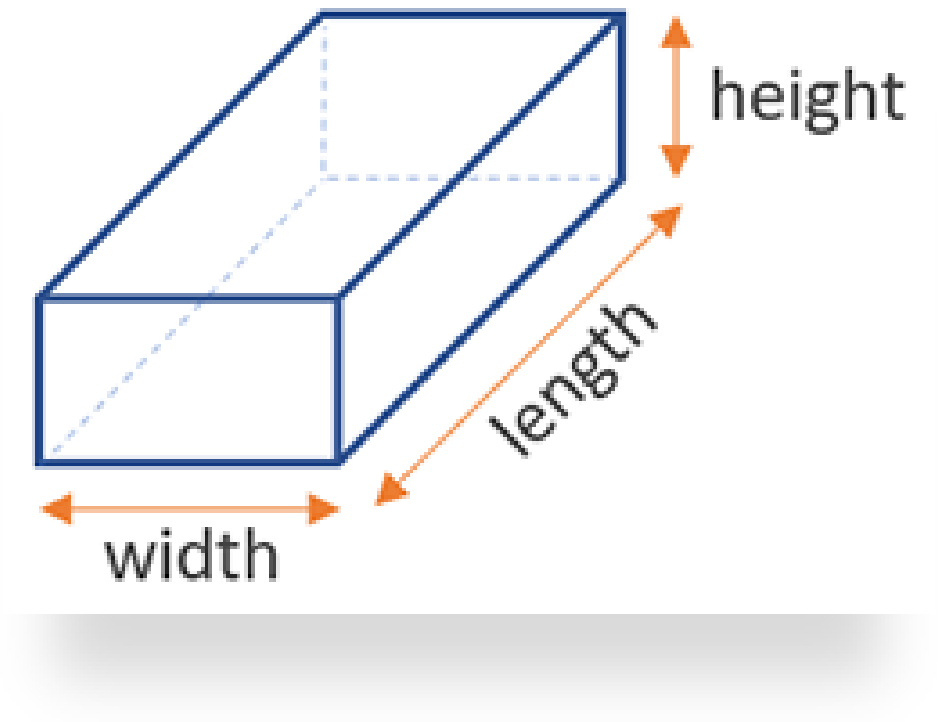
- ออบเจ็คตที่สร้างมาจากคลาสเดียวกันจะมีคุณสมบัติและความสามารถเหมือนกัน




More
Restrictive
Private
Default
Protected
Public
Less
Restrictive
JAVA
Access
Modifiers
tutorial.eyehunts.com


Access Modifiers
Default
public
protected
private
Visible to the package, the default. No modifiers are needed.
Visible to the world
Visible to the package and all subclasses
Visible to the class only

Ex. V1

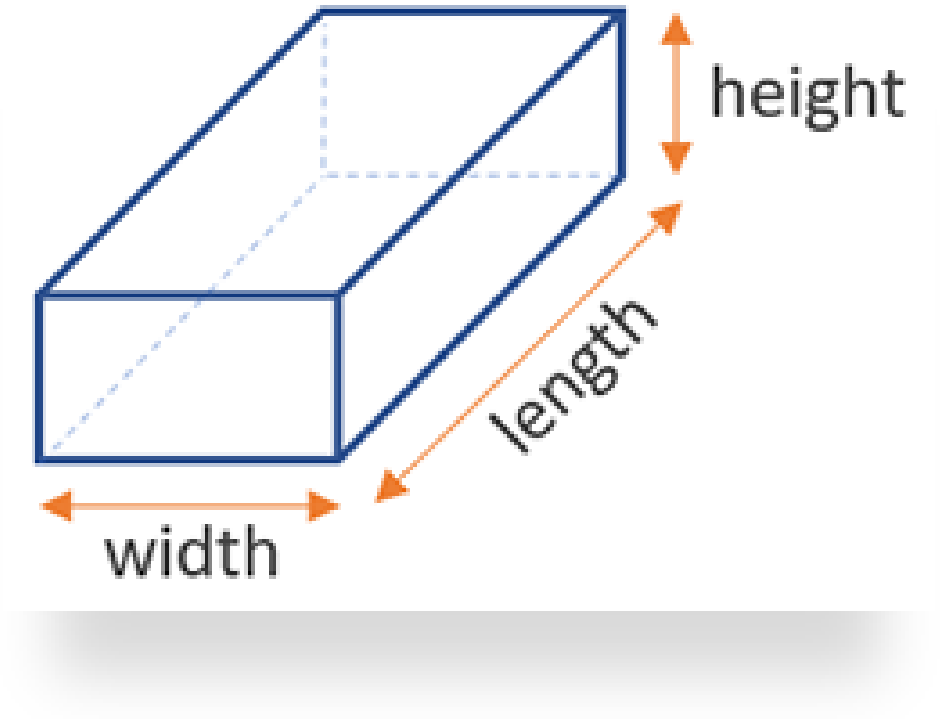



```
public class Box {
 public double width, length, height;

 public double volume(){
  return width * length * height;
 }

 public double surfaceArea(){
  return (2.0 * width * height) + (2.0 * width * length) + (2.0 * length * height);
 }
}
```

```
public class Main {
 public static void main(String[] args){
   Box aBox = new Box();
   aBox.width = 10.0;
   aBox.length = 5.0;
   aBox.height = 7.0;

   System.out.println(aBox.volume());
   System.out.println(aBox.surfaceArea());
 }
}
```

setter

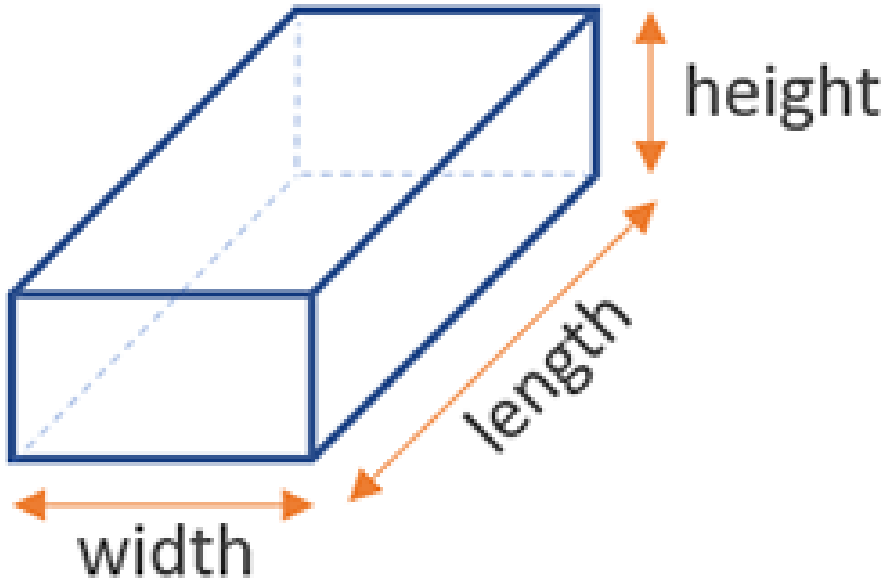
Ex. V2 -> setter



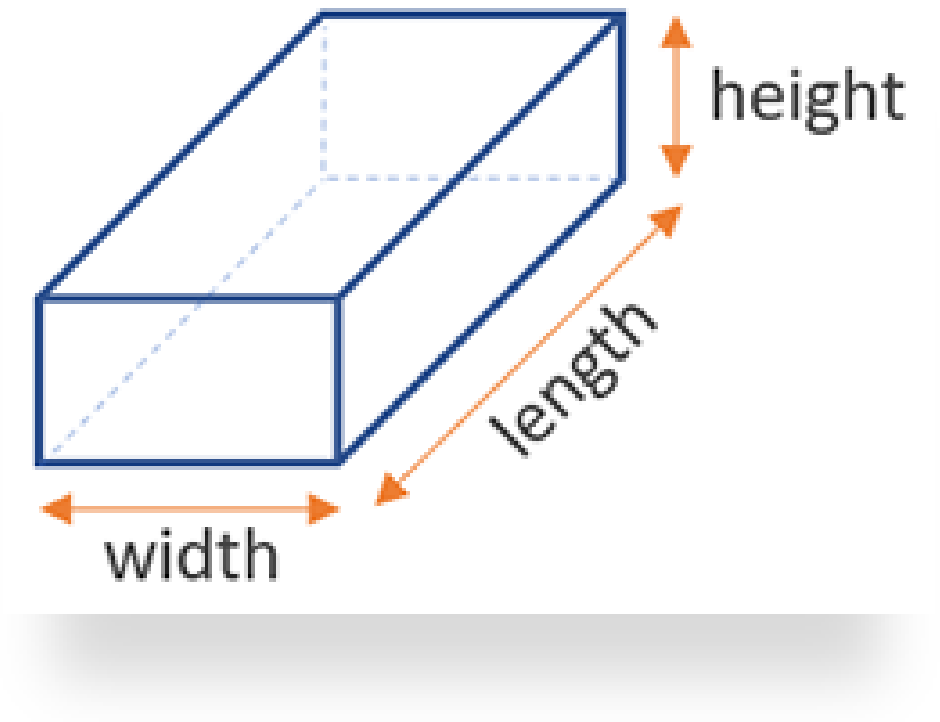

```
public class Box {
 private double width, length, height;

 //getter, setter
public void setW(double w){
  this.width  = w;
}

public void setL(double l){
  this.length = l;
}
```

Ex. V2 -> setter

```
public void setH(double h){
   this.height= h;
}
public double volume(){
 return width * length * height;
}

public double surfaceArea(){
 return (2.0 * width * height) + (2.0 * width * length) + (2.0 * length * height);
}
}
```



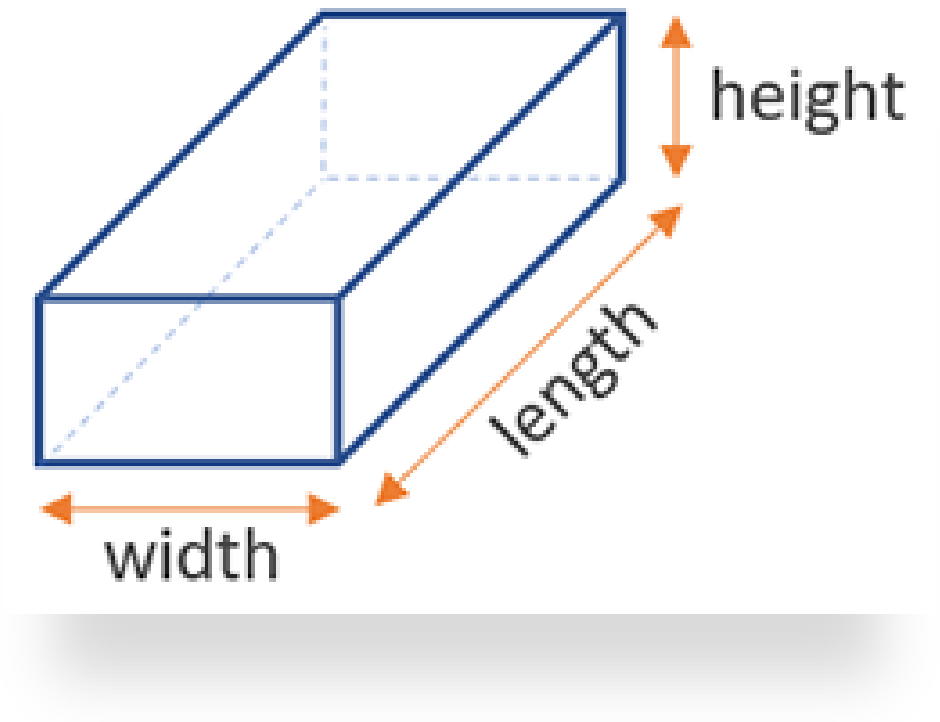
Ex. V2 -> setter




```
public class Main {
 public static void main(String[] args){
   Box aBox = new Box();
   aBox.setW(10.0);
   aBox.setL(5.0);
   aBox.setH(7.0);

   System.out.println(aBox.volume());
   System.out.println(aBox.surfaceArea());
 }
}
```

constructor

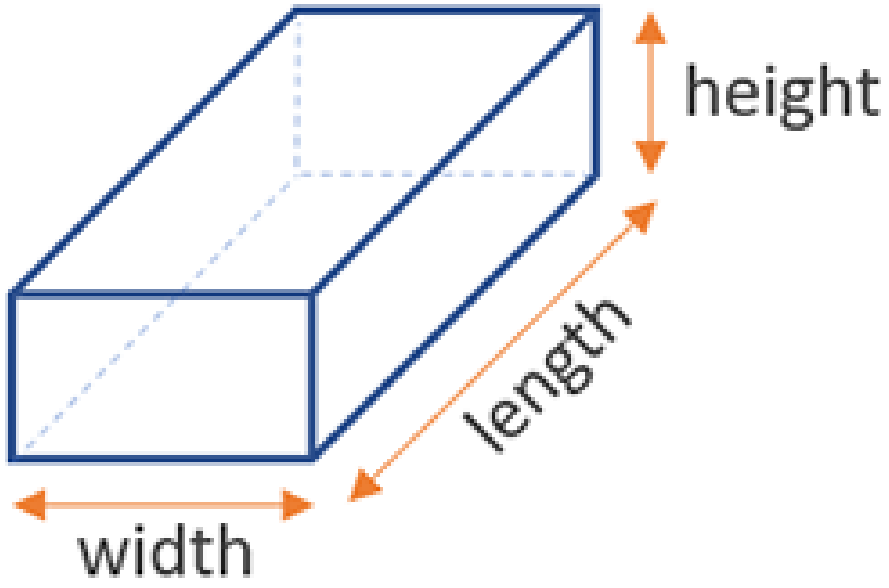
Ex. V3 -> constructor



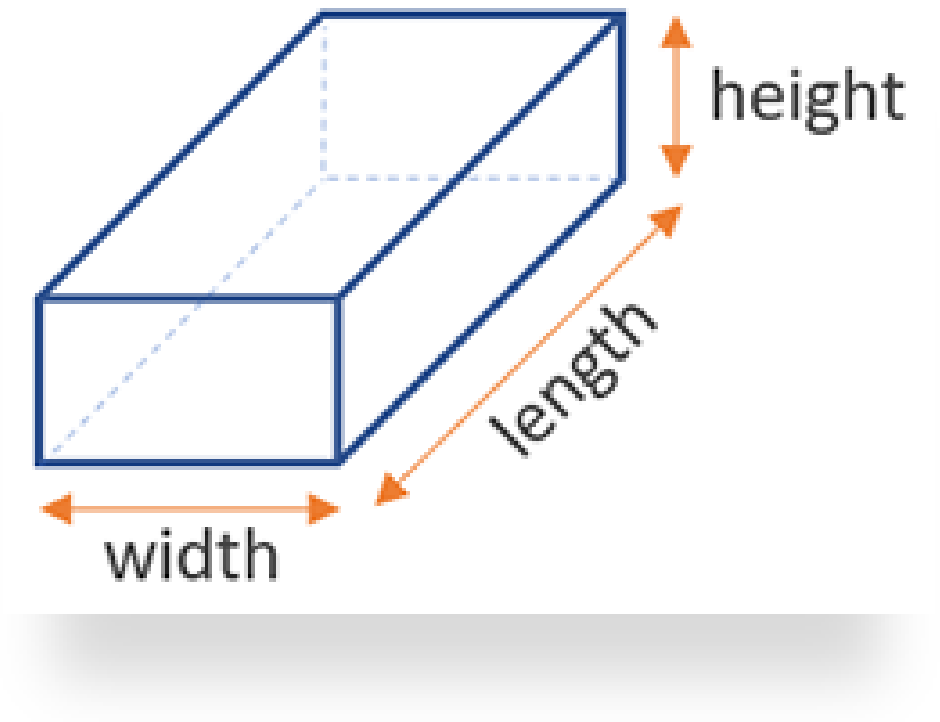

```
public class Box {
 private double width, length, height;

 //getter, setter
 //constructor
public Box(double w, l, h){
   setW(w);
   setL(l);
   setH(h);
}

public void setW(double w){
   this.width  = w;
}

public void setL(double l){
   this.length = l;
}
```

Ex. V3 -> constructor

```
public void setH(double h){
   this.height= h;
}
public double volume(){
 return width * length * height;
}

public double surfaceArea(){
 return (2.0 * width * height) + (2.0 * width * length) + (2.0 * length * height);
}
}
```



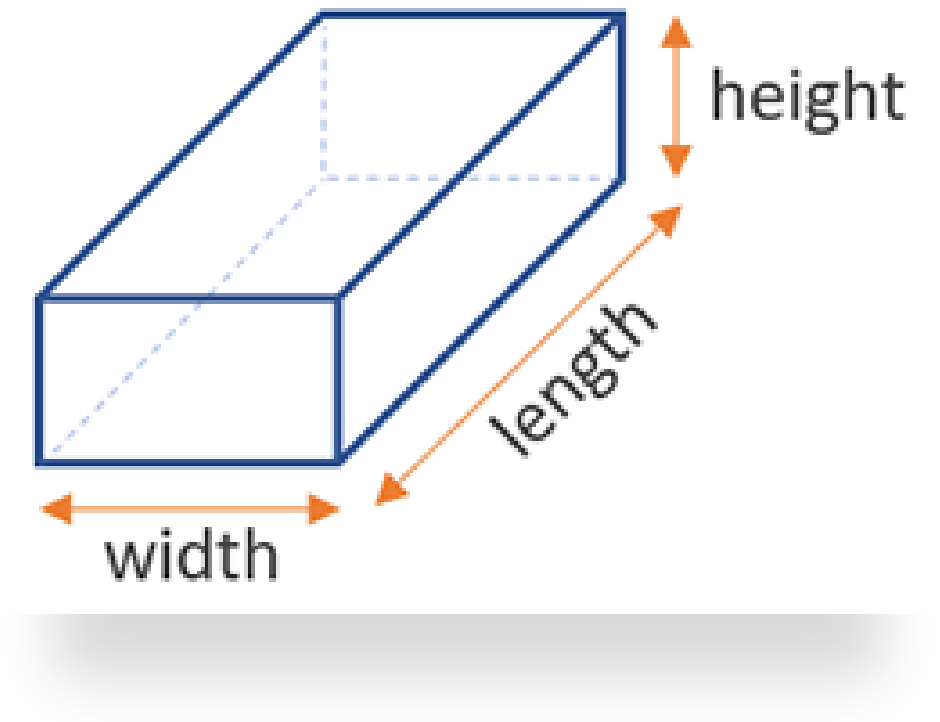
Ex. V3 -> constructor




```
public class Main {
 public static void main(String[] args){
   Box aBox = new Box(10,5,7);

   System.out.println(aBox.volume());
   System.out.println(aBox.surfaceArea());
 }
}
```

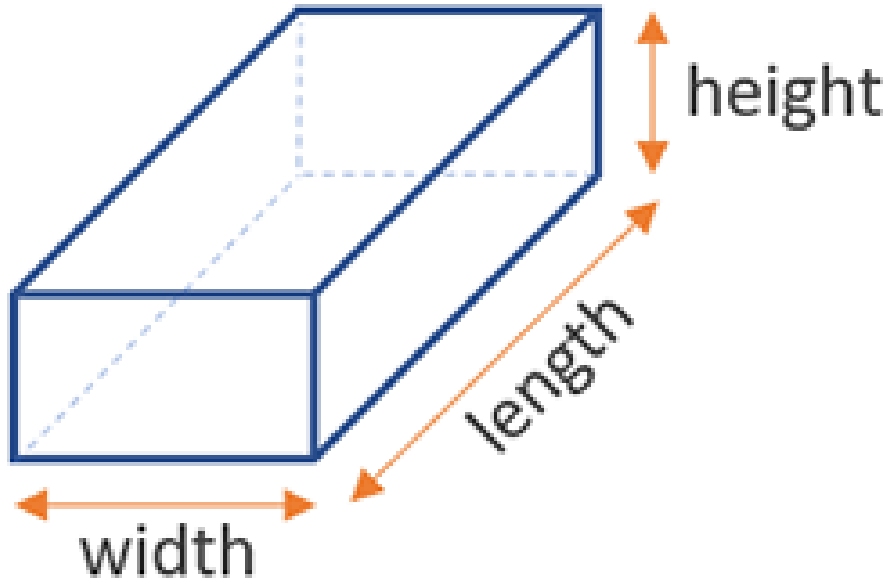
getter

Ex. V3 -> getter



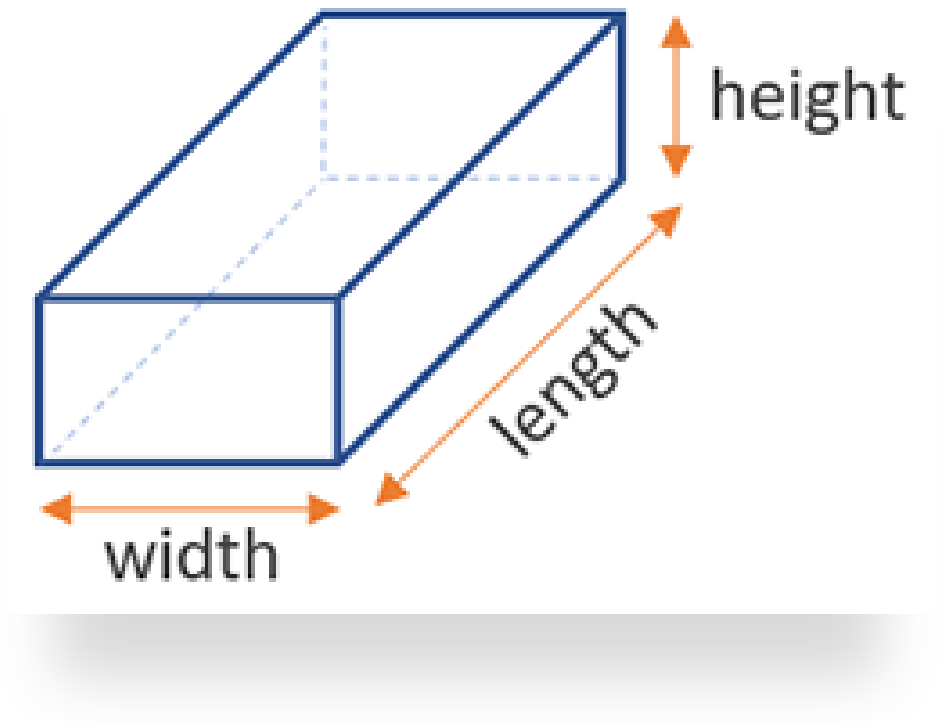

```
public class Box {
 private double width, length, height;

 //getter, setter
 //constructor
 public Box(double w, l, h){
   setW(w);
   setL(l);
   setH(h);
 }

 public void setW(double w){
   this.width  = w;
 }

 public void setL(double l){
   this.length = l;
 }
```

Ex. V3 -> getter




```
public double getW(){
  return w;
}
public void setH(double h){
   this.height= h;
 }
 public double volume(){
  return width * length * height;
 }

 public double surfaceArea(){
  return (2.0 * width * height) + (2.0 * width * length) + (2.0 * length * height);
 }
}
```

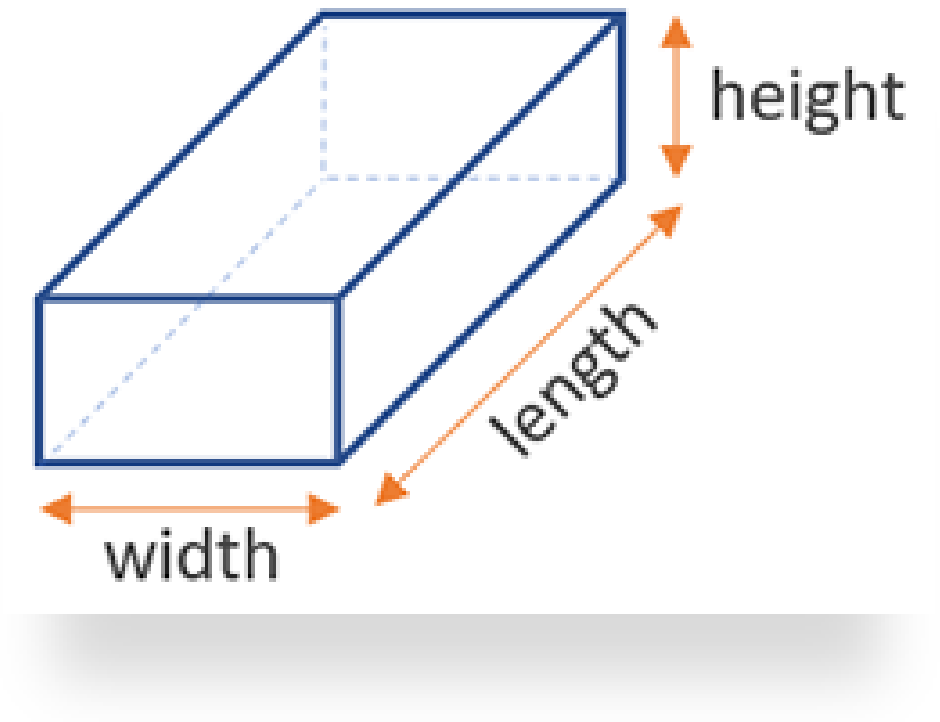
Ex. V3 -> getter




```
public class Main {
 public static void main(String[] args){
   Box aBox = new Box(10,5,7);

   System.out.println(aBox.volume());
   System.out.println(aBox.surfaceArea());

   if(aBox.getW() > 100.0){
     System.out.println("กล่องขนาดใหญ่");
   }
 }
}
```

annotation

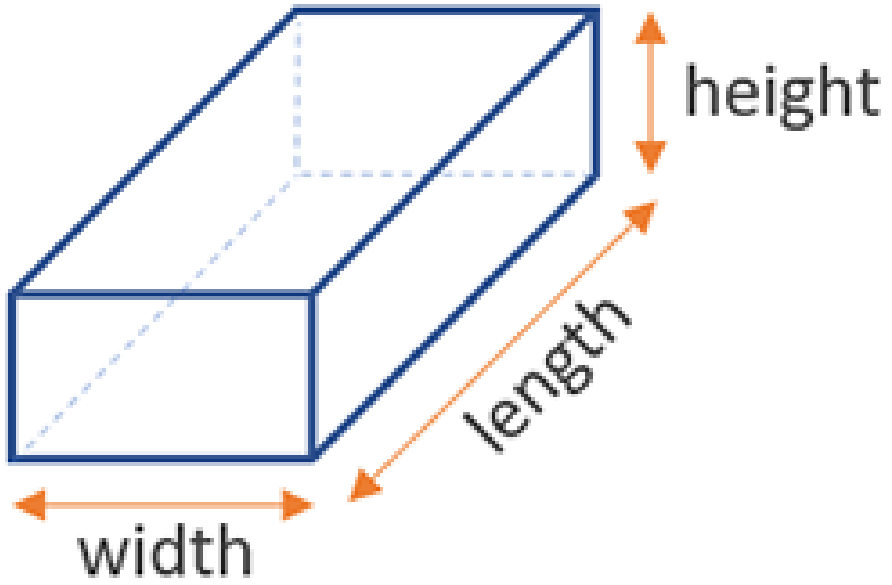
Ex. V4 -> annotation



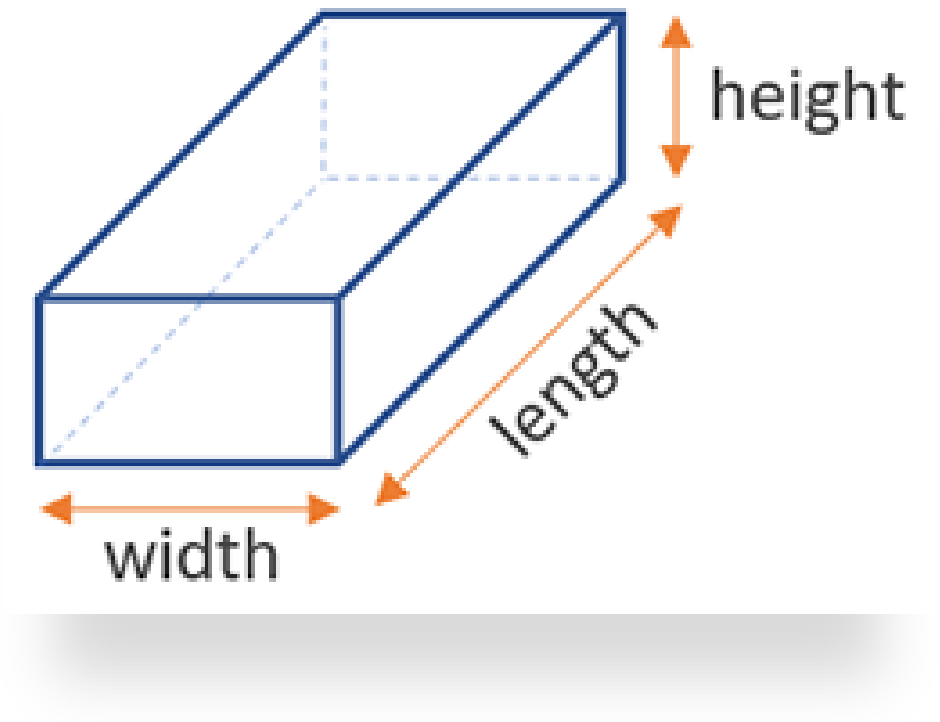

```
public class Box {
 private double width, length, height;

 //getter, setter
 //constructor
public Box(double w, l, h){
   setW(w);
   setL(l);
   setH(h);
}

public void setW(double w){
   this.width  = w;
}

public void setL(double l){
   this.length = l;
}
```

Ex. V4 -> annotation

```
public double getW(){
   return w;
}
public void setH(double h){
    this.height= h;
 }
 public double volume(){
  return width * length * height;
 }

 public double surfaceArea(){
  return (2.0 * width * height) + (2.0 * width * length) + (2.0 * length * height);
 }

 //annotation
@Override
  public String toString(){
     return String.format("width = %.2f length = %.2f height = %.2f :: Volume = %.2f ",width, length,
height, Volume());
}
```



Ex. V4 -> annotation




```
public class Main {
 public static void main(String[] args){
   Box aBox = new Box(10,5,7);

   System.out.println(aBox.volume());
   System.out.println(aBox.surfaceArea());

   if(aBox.getW() > 100.0){
      System.out.println("กล่องขนาดใหญ่");
   }
  System.out.println(aBox);
 }
}
```