



Smit IT Institute

Gives you wings to fly high

Access Modifiers	Public	Private	Protected	Default
Same Class	Yes	Yes	Yes	Yes
Same Package extended class(Sub-Class)	Yes	No	Yes	Yes
Same Package Non-Sub Class. i.e Not Extended Class	Yes	No	Yes	Yes
Different Package Extended Class (Sub-Class)	Yes	No	Yes	No
Different Package Non-Sub Class. i.e Not Extended Class	Yes	No	No * To make it Yes Inheritance is Required.	No



Encapsulation

1. It is a process of Logically Binding two or more different Logical Entities such as class with each other with respect to its Methods and Data Members in order to Store, Access or Retrieve Data as required.
2. It is an Object Oriented Concepts.
3. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.



Package 1

```
package Package1;
```

```
class MyClass  
{  
    String a="Default";  
    public String pub="Public";  
    protected String pro="protected";  
    private String pr="private";  
}
```

```
public class Pack1Main1 extends MyClass  
{  
  
    public static void main(String[] args)  
    {  
        MyClass my=new MyClass();  
        System.out.println("Default a="+my.a);  
        System.out.println("Public pub="+my.pub);  
        System.out.println("Protected pro="+my.pro);  
        System.out.println("Private pr="+my.pr);→ERROR  
(Private Member Cannot be Accessed Outside Its Class)  
    }  
}
```

Output:-

```
Default a=Default  
Public pub=Public  
Protected pro=protected
```



Package 1

```
package Package1;

public class Pack1Main2
{
    public static void main(String[] args)
    {
        Pack1Main1 pm1=new Pack1Main1();
        //Pack1Main1 Class Binded With Pack1Main2.(Encapsulation)

        System.out.println("Default a="+pm1.a);
        System.out.println("Public pub="+pm1.pub);
        System.out.println("Protected pro="+pm1.pro);
        System.out.println("Private pr="+pm1.pr);→ERROR
        (Private Member Cannot be Accessed Outside Its Class)

    }
}
```

Output:-

```
Default a=Default
Public pub=Public
Protected pro=protected
```



Package 2

```
package Package2;
import Package1.Pack1Main1;

public class Pack2Main3 extends Pack1Main1
{
    public static void main(String[] args)
    {
        Pack1Main1 pm1=new Pack1Main1();
        Pack2Main3 pm3=new Pack2Main3();->Object Of
Current Main Class

        // System.out.println("Default a="+pm1.a);
        System.out.println("Public pub="+pm1.pub);
        System.out.println("Protected pro="+pm3.pro);
    }
}
```

Note:- To Access Protected members from Other Package, Inheritance is required and Object of Current Main Class Should be Created.

Output:-

Public pub=Public
Protected pro=protected



Encapsulation Example:

Step 1:

Generating Getters and Setters

1. Create a Class and Declare Data Members.
2. Eg: int id,String name, long salary;
3. Remove or Do not Mention Main Function.
4. Select all members.

```
1 package Package3;  
2  
3 class Bean2  
4 {  
5  
6  
7     int id;  
8     String name;  
9     long salary;  
10  
11 }  
12
```



Smit IT Institute
Gives you wings to fly high

Step 2: Right Click for Source.

The screenshot shows an IDE interface with a project explorer on the left, a code editor in the center, and a context menu on the right. The project explorer shows a hierarchy: Encapsulation > JRE System Library [JavaSE-17] > src > Package1 > Package2 > Package3 > Bean2.java. The code editor displays the following code:

```
1 package Package3;  
2  
3 class Bean2  
4 {  
5  
6  
7     int id;  
8     String name;  
9     long salary;  
10  
11 }  
12
```

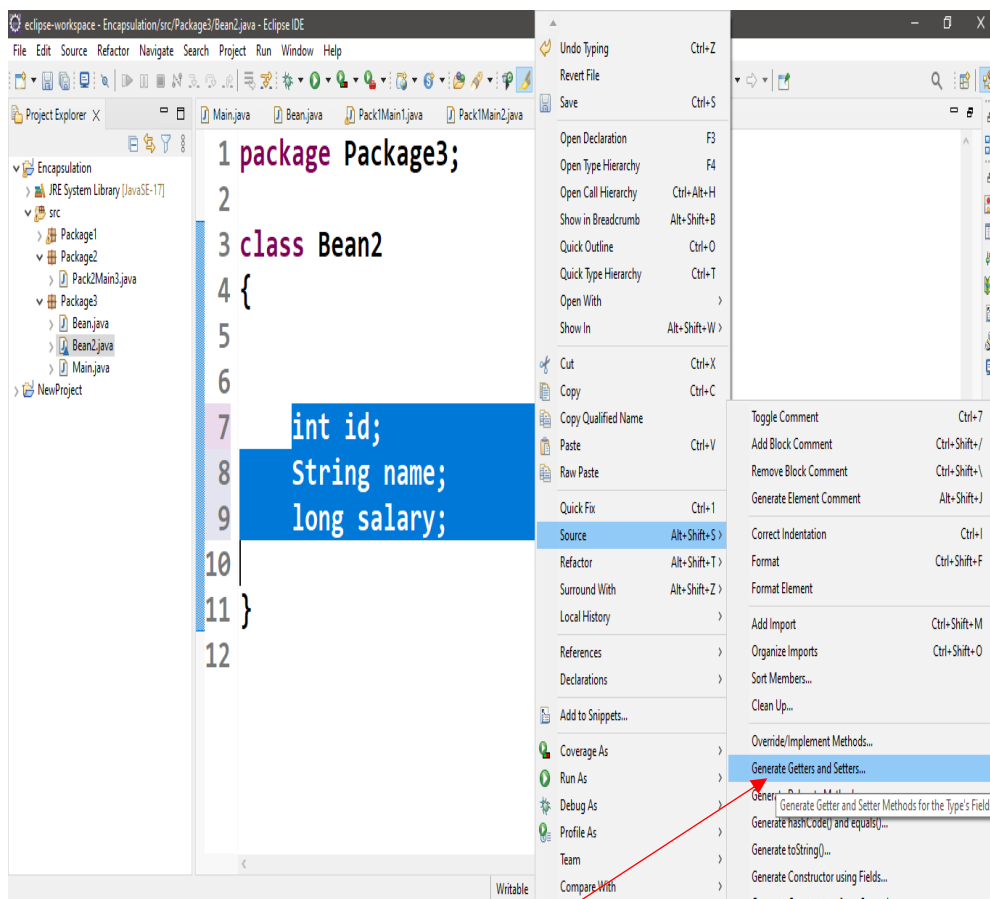
The context menu is open, showing various actions. The 'Source' option is highlighted, and a red arrow points to it. The menu items include:

- Undo Typing (Ctrl+Z)
- Revert File
- Save (Ctrl+S)
- Open Declaration (F3)
- Open Type Hierarchy (F4)
- Open Call Hierarchy (Ctrl+Alt+H)
- Show in Breadcrumb (Alt+Shift+B)
- Quick Outline (Ctrl+O)
- Quick Type Hierarchy (Ctrl+T)
- Open With >
- Show In (Alt+Shift+W >)
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Copy Qualified Name
- Paste (Ctrl+V)
- Raw Paste
- Quick Fix (Ctrl+I)
- Source (Alt+Shift+S >)
- Refactor (Alt+Shift+T >)
- Surround With (Alt+Shift+Z >)
- Local History >
- References >
- Declarations >
- Add to Snippets...
- Coverage As >
- Run As >
- Debug As >
- Profile As >
- Team >
- Compare With >



Smit IT Institute
Gives you wings to fly high

Step 3: Go For generate getters and setters.

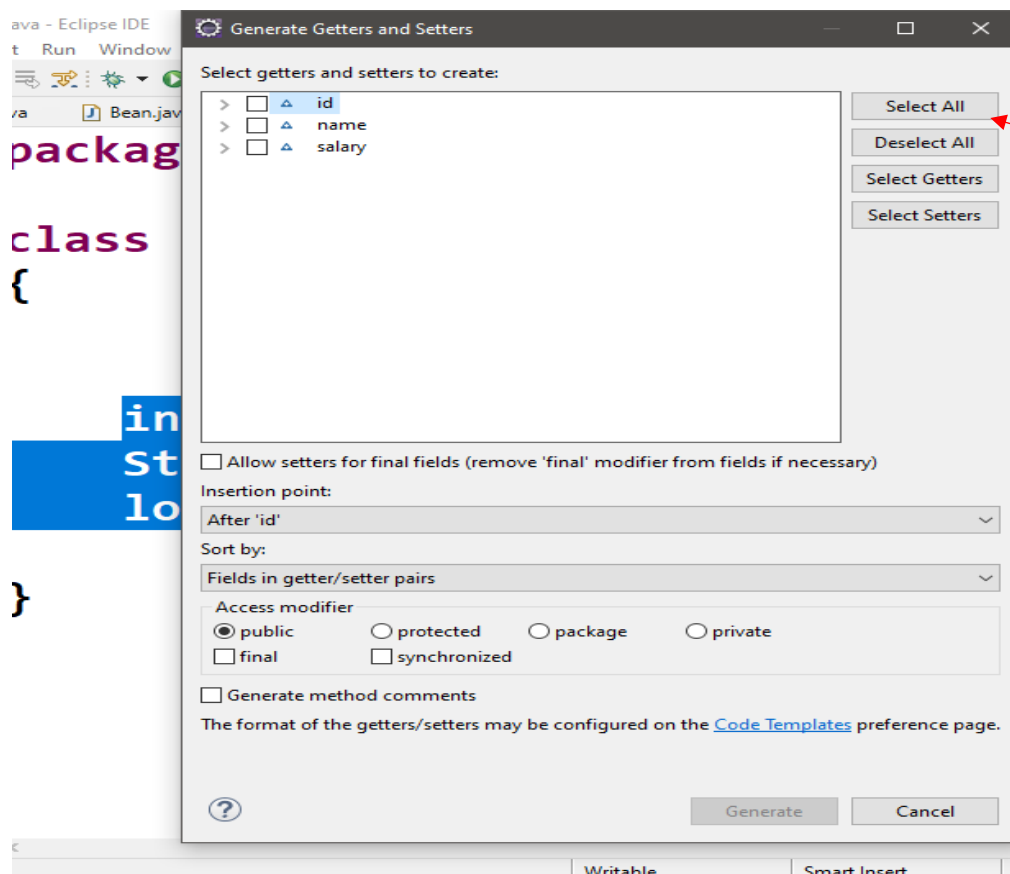




Smit IT Institute

Gives you wings to fly high

Step 4: Click Select All.

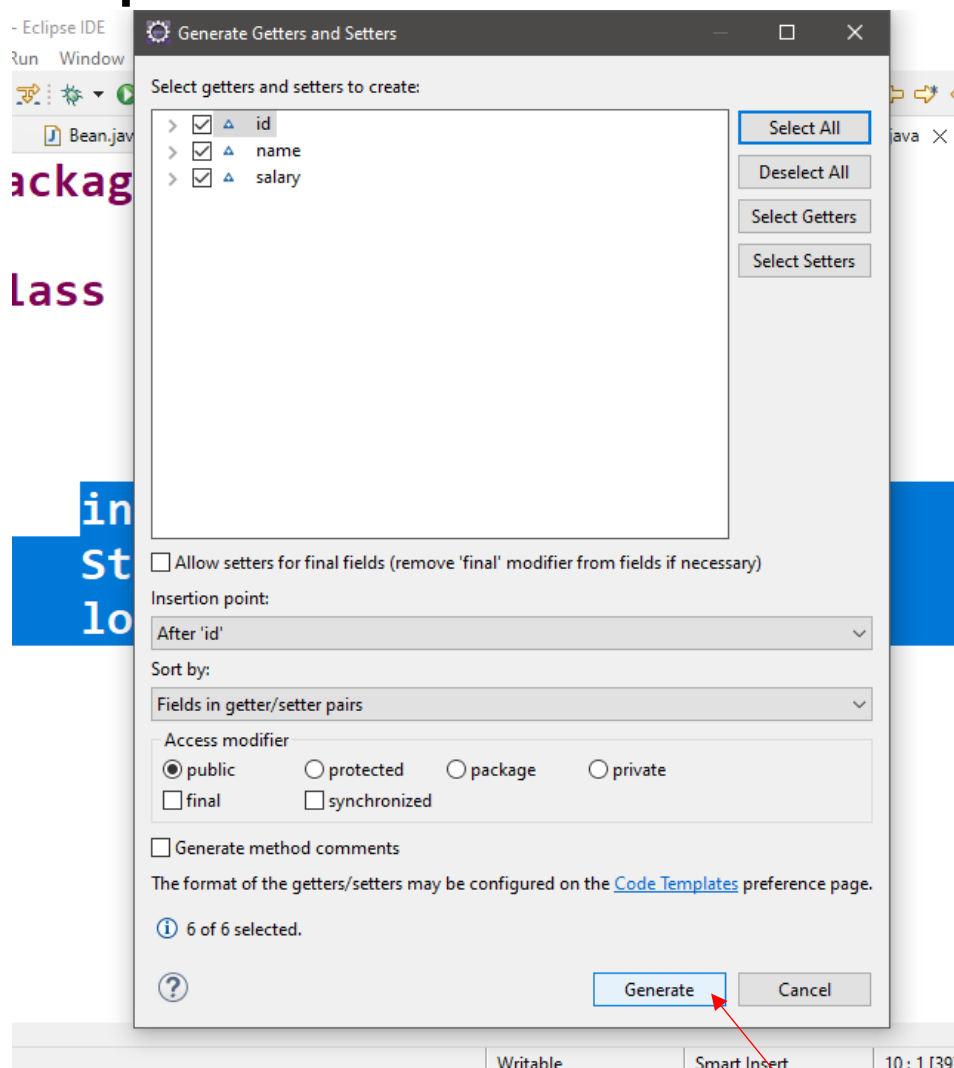




Smit IT Institute

Gives you wings to fly high

Step 5: Click On Generate





Smit IT Institute
Gives you wings to fly high

We Get the Following Output:

```
pse-workspace - Encapsulation/src/Package3/Bean2.java - Eclipse IDE
Edit Source Refactor Navigate Search Project Run Window Help
Main.java Bean.java Pack1Main1.java Pack1Main2.java Pack2Main3.java Bean2.java X
1 package Package3;
2
3 class Bean2
4 {
5     int id;
6     String name;
7     long salary;
8     public int getId() {
9         return id;
10    }
11    public void setId(int id) {
12        this.id = id;
13    }
14    public String getName() {
15        return name;
16    }
17    public void setName(String name) {
18        this.name = name;
19    }
20    public long getSalary() {
21        return salary;
22    }
23    public void setSalary(long salary) {
24        this.salary = salary;
25    }
26 }
```

Writable Smart Insert 9:17:110



```
package Package3;

public class Bean2
{

    int id;
    String name;//Rohit
    long salary;

    public int getId()
    {
        return id;
    }
    public void setId(int id)
    {
        this.id = id;
    }
    public String getName()
```



```
{  
    return name;//Rohit  
}  
  
public void setName(String name)//Rohit  
{  
    this.name = name;  
}  
public long getSalary()  
{  
    return salary;  
}  
public void setSalary(long salary)  
{  
    this.salary = salary;  
}  
}
```



Encapsulated By Main Class In Same Package i.e Package3.

```
package Package3;
```

```
public class Main  
{
```

```
    public static void main(String[] args)  
    {  
        Bean2 b=new Bean2();  
        b.setId(1);  
        System.out.println("Id is:"+b.getId());  
  
        b.setName("Rohit");  
        System.out.println("Name is:"+b.getName());  
  
        b.setSalary(5000);  
        System.out.println("Salary is:"+b.getSalary());  
    }
```

```
}
```

Output:-



Smit IT Institute

Gives you wings to fly high

Id is:1

Name is: Rohit

Salary is:5000