Access Specifiers-

Access specifier's plays very important role while performing the operation on variable, methods, classes, etc.

In other words, it is simply used to restrict the access.

There are four types of access specifiers as

Private

Default

Protected

Public

Private-

It apply to global variable, method, constructor and inner class only.

Class cannot be private.

It can access within class only not outside class or outside package as scope is very limited.

Local variables cannot private.

Default-

It apply to global variable, local variable, constructors, method, inner class and outer class.

It can be accessible within the same package only.

When the access specified is not specified then it will be treated as default members.

No need to use keyword default like private.

Protected-

It apply to constructor, global variables, inner class and methods.

It cannot apply to local variables and outer class.

It is accessible within the same package and also possible into another package if inheritance is happened while calling.

Public-

It apply to class, method, constructor, global variable, static variable, inner class, outer class.

It can access anywhere in the class or outside the class or same package or different package

Local variables cannot public because they have limited scope within the method only. If we make it public then getting error only. "Illegal modifier for parameter"

Note-

1. We can apply default access Specifiers or final on local variable.

	Private	Default	Protected	public
Same class	Yes	Yes	Yes	Yes
Same package sub-class	No	Yes	Yes	Yes
Same package non-subclass	No	Yes	Yes	Yes
Different package sub-class	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

Why we use access specifiers?

If we have a business requirement where we need to perform the employee CRUD operations and all the methods need to be called from `getEmployeeData()` only.

Scenario-1

```
package com.wipro.velocity;
```

```
public class Employee {

    public void addOperation() {

        System.out.println("Add operation");
    }

    public void editOperation() {
        System.out.println("Edit operation");
    }

    public void getOperation() {
        System.out.println("Get operation");
    }

    public void deleteOperation() {
        System.out.println("Delete operation");
    }

    public void getEmployeeData() {
        addOperation();
        editOperation();
        getOperation();
        deleteOperation();
    }
}
```

Here we are directly call any method from outside class because scope is public. Hence requirement is not fulfilled here.

Scenario-2

class Employee{

```
package com.wipro.velocity;

public class Employee {
```

```java
    private void addOperation() {

        System.out.println("Add operation");
    }

    private void editOperation() {
        System.out.println("Edit operation");
    }

    private void getOperation() {
        System.out.println("Get operation");
    }

    private void deleteOperation() {
        System.out.println("Delete operation");
    }

    public void getEmployeeData() {
        addOperation();
        editOperation();
        getOperation();
        deleteOperation();
    }
}
```

Here we cannot directly call any method except getEmployeeData() because scope is private. So it cannot be directly accessible from outside. We need to access it from by calling `getEmployeeData` ().

In this way, we use the access specifiers.