

# Public vs Protected vs Package vs Private Access Modifier in Java

Last Updated : 08 Oct, 2021

Whenever we are writing our classes we have to provide some information about our classes to the JVM like whether this class can be accessible from anywhere or not, whether child class creation is possible or not, whether object creation is possible or not etc. we can specify this information by using an appropriate keyword in java called access modifiers. So access modifiers are used to set accessibility of classes, methods, and other members.

## Modifier 1: Public Access Modifiers

If a class is declared as public then we can access that class from anywhere.

In the below example we are creating a package pack1 inside that package we declare a class A which is public and inside that class, we declare a method m1 which is also public. Now we create another package pack2 and inside that pack2 we import pack1 and declare a class B and in class B's main method we create an object of type class A and trying to access the data of method m1.

### Example:

---

```
// Java program to showcase the example
// of public access modifier

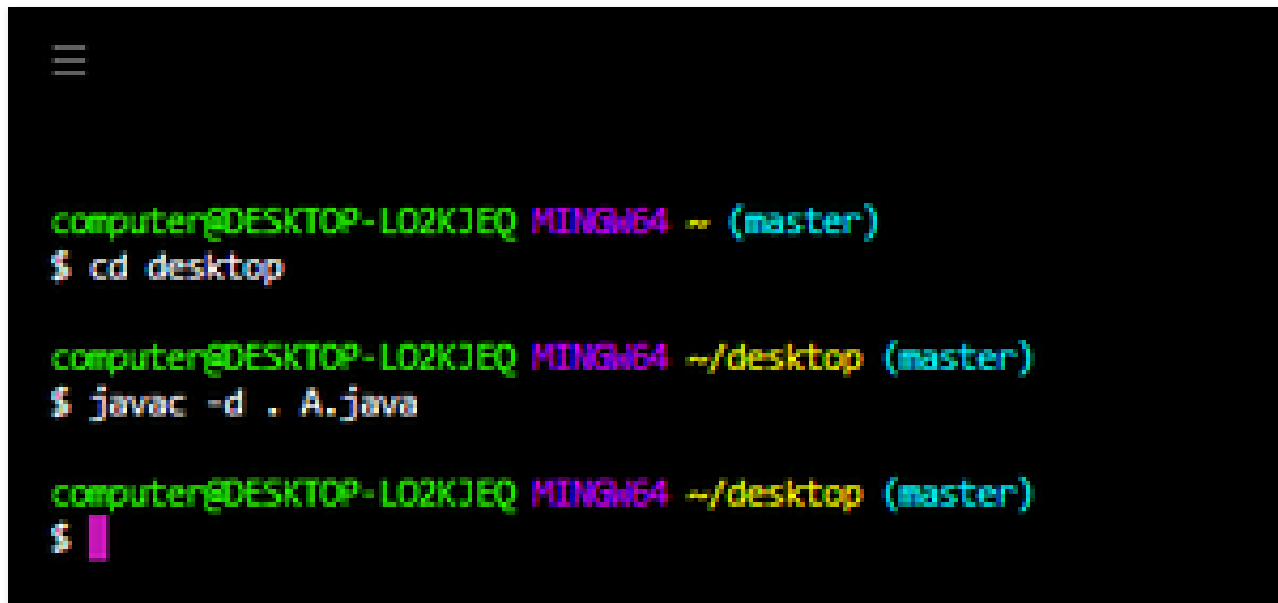
// creating a package
package pack1;

// import required packages
import java.io.*;
import java.util.*;

// declaring a public class
public class A {

    // declaring method m1
    public void m1() { System.out.println("GFG"); }
}
```

Compiling and saving the above code by using the below command line:

A terminal window with a black background and green text. It shows three lines of commands being executed in a Windows command prompt. The first line shows the user navigating to the desktop directory. The second line shows the user compiling a Java file named A.java. The third line shows the user at the prompt, ready for the next command.

```
computer@DESKTOP-LO2KJEQ MINGW64 ~ (master)
$ cd desktop

computer@DESKTOP-LO2KJEQ MINGW64 ~/desktop (master)
$ javac -d . A.java

computer@DESKTOP-LO2KJEQ MINGW64 ~/desktop (master)
$
```

```
// creating a package
package pack2;

// import required packages
import java.io.*;
import java.util.*;

// importing package pack1
import pack1.A;

// driver class
class B {

    // main method
    public static void main(String[] args)
    {
        // creating an object of type class A
        A a = new A();

        // accessing the method m1()
        a.m1();
    }
}
```

```
computer@DESKTOP-L02KJ1EQ MINGW64 ~/desktop (master)
$ javac -d . A.java

computer@DESKTOP-L02KJ1EQ MINGW64 ~/desktop (master)
$ javac -d . B.java

computer@DESKTOP-L02KJ1EQ MINGW64 ~/desktop (master)
$ java pack2.B
GFG

computer@DESKTOP-L02KJ1EQ MINGW64 ~/desktop (master)
$
```

If class A is not public while compiling B class we will get a compile-time error saying pack1. A is not public in pack1 and can't be accessed from the outside package.

Similarly, a member or method, or interface is declared as public as we can access that member from anywhere.

## Modifier 2: Protected Access Modifier

This modifier can be applied to the data member, method, and constructor, but this modifier can't be applied to the top-level classes and interface.

A member is declared as protected as we can access that member only within the current package but only in the child class of the outside package.

## Implementation:

### Example

---

```
// Java program to showcase the example
// of protected access modifier
// import required packages
import java.io.*;
import java.util.*;

// declaring a parent class A
class A {

    // declaring a protected method m1()
    protected void m1() { System.out.println("GFG"); }
```

```
}

// creating a child class by extending the class A
class B extends A {

    // main method
    public static void main(String[] args)
    {
        // creating an object of parent class
        // using parent reference
        A a = new A();

        /// calling method m1
        a.m1();

        // creating an object of child class
        // using child reference
        B b = new B();

        // calling method m1
        b.m1();

        // creating an object of child class
        // using parent reference
        A a1 = new B();

        // calling m1 method
        a1.m1();
    }
}
```

## Output

GFG

GFG

GFG

## Output explanation:

In the above example, we create three objects using parent reference and child reference and call m1() method on it, and it successfully executed so from the above example we can say that we can access the protected method within the current package anywhere either by using parent reference or by child reference.

## Modifier 3: Private Access Modifiers

This modifier is not applicable for top-level classes or interfaces. It is only applicable to constructors, methods, and fields inside the classes.

If a variable or methods or constructor is declared as private then we can access them only from within the class i.e from outside the class we can't access them.

### Example:

---

```
// Java program to showcase the example
// of private access modifier

// import required packages
import java.io.*;

import java.util.*;

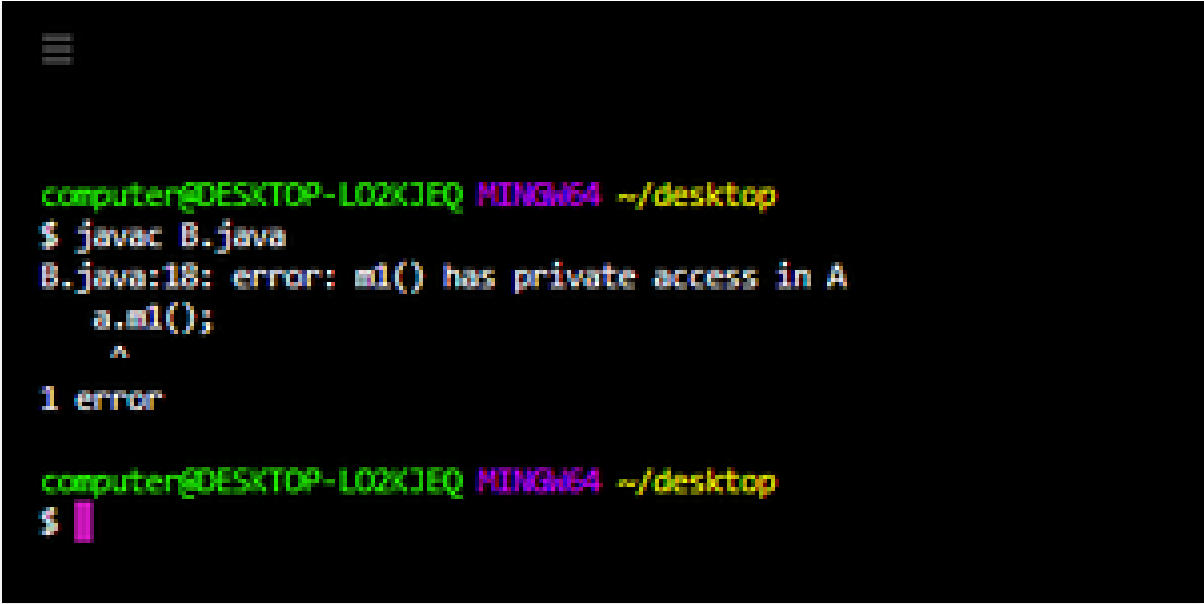
// helper class
class A {

    // helper method
    private void m1() { System.out.println("GFG"); }
}

// driver class
class B {

    // main method
    public static void main(String[] args)
    {
        // creating an object of type class A
        A a = new A();

        // accessing the method m1()
        a.m1();
    }
}
```



```
computer@DESKTOP-L02XJEQ MINGW64 ~/desktop
$ javac B.java
B.java:18: error: m1() has private access in A
    a.m1();
    ^
1 error

computer@DESKTOP-L02XJEQ MINGW64 ~/desktop
$
```

#### Modifier 4: Package(Default) Access Modifier

A class or method or variable declare without any access modifier then is considered that it has a package(default)access modifier The default modifier act as public within the same package and acts as private outside the package. If a class is declared as default then we can access that class only within the current package i.e from the outside package we can't access it. Hence, the default access modifier is also known as the package-level access modifier. A similar rule also applies for variables and methods in java.

#### Example:

---

```
// Java Program to illustrate Package Level Access Modifier

// Importing utility classes
// Importing input output classes
import java.io.*;
import java.util.*;

// Main Class
class GFG {

    // Declaring default variables that is
    // having no access modifier
    String s = "Geeksfor";
    String s1 = "Geeks";

    // Method 1
    // To declare a default method
    String fullName()
```

```

{

    // Concatenation of strings
    return s + s1;
}

// Method 2
// Main driver method
public static void main(String[] args)
{

    // Creating an object of main class(GFG)
    // in the main() method
    GFG g = new GFG();

    // Calling method1 using class instance
    // and printing the concation of strings
    System.out.println(g.fullName());
}
}

```

## Output

GeeksforGeeks

Finally, after getting it done with all four access modifiers let us conclude the evident differences between them

Public Access Modifier	Private Access Modifier	Protected access modifier	Package access modifier
This modifier is applicable for both top-level classes and interfaces.	This modifier is not applicable for both top-level classes and interfaces.	This modifier is not applicable for both top-level classes and interfaces.	This modifier is applicable for both top-level classes and interfaces.
Public members can be accessed from the child class of the same package.	Private members cannot be accessed from the child class of the same package.	Protected members can be accessed from the child class of the same package.	Package members can be accessed from the child class of the same package.

Public Access Modifier	Private Access Modifier	Protected access modifier	Package access modifier
Public member can be accessed from non-child classes of the same package.	Private members cannot be accessed from non-child classes of the same package.	Protected member can be accessed from non-child classes of the same package.	Package member can be accessed from non-child class of the same package.
Public members can be accessed from the child class of outside package.	Private members cannot be accessed from the child class of outside package.	Protected members can be accessed from the child class of the outside package, but we should use child reference only.	Package members cannot be accessed from the child class of outside package.
Public members can be accessed from non-child class of outside package.	Private members cannot be accessed from non-child class of outside package.	Protected members cannot be accessed from the non-child class of outside package.	Package members cannot be accessed from non-child class of outside package.
Public modifier is the most accessible modifier among all modifiers.	Private modifier is the most restricted modifier among all modifiers.	Protected modifier is more accessible than the package and private modifier but less accessible than public modifier.	Package modifier is more restricted than the public and protected modifier but less restricted than the private modifier.

## Only Java Can Get The Job Done For You.

So Strengthen Your Foundations and

**Start Learning**

