

# **Programming Language II**

## **CSE-215**

Prof. Dr. Mohammad Abu Yousuf  
yousuf@juniv.edu

# Access Modifiers in java

- The access modifiers in java specifies accessibility (scope) of a data member, method, constructor or class.
- There are 4 types of java access modifiers:
  - private
  - default
  - protected
  - public

# private access modifier

- The private access modifier is accessible only within class.

```
1.class A{
2.private int data=40;
3.private void msg(){System.out.println("Hello java");}
4.}
5.
6.public class Simple{
7. public static void main(String args[]){
8.   A obj=new A();
9.   System.out.println(obj.data);//Compile Time Error
10.  obj.msg();//Compile Time Error
11.  }
12.}
```

# private access modifier

- Role of Private Constructor
  - If you make any class constructor private, you cannot create the instance of that class from outside the class. For example:

```
1.class A{  
2.private A(){}//private constructor  
3.void msg(){System.out.println("Hello java");}  
4.}  
5.public class Simple{  
6. public static void main(String args[]){  
7.   A obj=new A();//Compile Time Error  
8. }  
9. }
```

Note: A class cannot be private or protected except nested class.

# default access modifier

- If you don't use any modifier, it is treated as **default** by default. The default modifier is accessible only within package.

```
1.//save by A.java
2.package pack;
3.class A{
4.  void msg(){System.out.println("Hello");}
5.}
```

```
1.//save by B.java
2.package mypack;
3.import pack.*;
4.class B{
5.  public static void main(String args[]){
6.    A obj = new A();//Compile Time Error
7.    obj.msg();//Compile Time Error
8.  }
9.}
```

the scope of class A and its method msg() is default so it cannot be accessed from outside the package.

# protected access modifier

- The **protected access modifier** is accessible within package and outside the package but through inheritance only.
- The protected access modifier can be applied on the data member, method and constructor. It can't be applied on the class.

```
1.//save by A.java  
2.package pack;  
3.public class A{  
4.protected void msg(){System.out.println("Hello");}  
5.}
```

```
1.//save by B.java  
2.package mypack;  
3.import pack.*;  
4.class B extends A{  
5. public static void main(String args[]){  
6.   B obj = new B();  
7.   obj.msg();  
8. }  
9.}
```

Output: Hello

# public access modifier

- The **public access modifier** is accessible everywhere. It has the widest scope among all other modifiers.

```
1.//save by A.java
2.
3.package pack;
4.public class A{
5.public void msg(){System.out.println("Hello");}
6.}
```

```
1.//save by B.java
2.
3.package mypack;
4.import pack.*;
5.class B{
6. public static void main(String args[]){
7.   A obj = new A();
8.   obj.msg();
9. }
10.}
```

Output: Hello



## All access modifiers

Access Modifier	within class	within package	outside package by subclass only	outside package
<b>Private</b>	Y	N	N	N
<b>Default</b>	Y	Y	N	N
<b>Protected</b>	Y	Y	Y	N
<b>Public</b>	Y	Y	Y	Y

# Java access modifiers with method overriding

- If you are overriding any method, overridden method (i.e. declared in subclass) must not be more restrictive.

```
1.class A{
2.protected void msg(){System.out.println("Hello java");}
3.}
4.
5.public class Simple extends A{
6.void msg(){System.out.println("Hello java");} //C.T.Error because msg() is default
7. public static void main(String args[]){
8.     Simple obj=new Simple();
9.     obj.msg();
10. }
11.}
```

Thank you