# SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

## An Autonomous Institution

Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

23CSB101

# OBJECT ORIENTED PROGRAMMING

## PACKAGES

By

M.Kanchana

Assistant Professor/CSE

# Access Specifiers

Access specifiers are used to specify the visibility and accessibility of a class constructors, member variables and methods.

Types:

1. Public
2. Private
3. Protected
4. Default (package)

# Access Specifiers

**Public (anything declared as public can be accessed from anywhere):**

A variable or method declared/defined with the public modifier can be accessed anywhere in the program through its class objects, through its subclass objects and through the objects of classes of other packages also.

**Private (anything declared as private can't be seen outside of the class):**

The instance variable or instance methods declared/initialized as private can be accessed only by its class. Even its subclass is not able to access the private members.

# Access Specifiers

**Protected (anything declared as protected can be accessed by classes in the same package and subclasses in the other packages):**

The protected access specifier makes the instance variables and instance methods visible to all the classes, subclasses of that package and subclasses of other packages.

**Default (can be accessed only by the classes in the same package):**

The default access modifier is friendly. This is similar to public modifier except only the classes belonging to a particular package knows the variables and methods.

# Access Specifiers

| | PRIVATE | DEFAULT | PROTECTED | PUBLIC |
|---|---|---|---|---|
| Same class | Yes | Yes | Yes | Yes |
| Same package Subclass | No | Yes | Yes | Yes |
| Same package Non-subclass | No | Yes | Yes | Yes |
| Different package Subclass | No | No | Yes | Yes |
| Different package Non-subclass | No | No | No | Yes |

# Packages and Access Specifiers

## Creating Own Package

- Choose a package name (e.g., mypackage1).
- Add the **package keyword** at the top of the Java file.
- Save the file inside a folder with the same name as the package.

package **packagename;**

```
package mypackage1;
public class FirstClass {
    public String i = "I am public variable";
    protected String j = "I am protected variable";
    private String k = "I am private variable";
    String r = "I don't have any modifier";
}
```

# Packages and Access Specifiers

**Compilation Command:**

**javac -d . mypackage1\FirstClass.java**

-d . → Saves the compiled .class file inside the package directory mypackage1.

D:\JAVA Programs\
├── mypackage1\
│ ├── FirstClass.class

**Run:**

**java mypackage1.FirstClass**

# Packages and Access Specifiers

```
package mypackage2;
import mypackage1.FirstClass;

class SecondClass extends FirstClass
{
    void method() {
System.out.println(i);
System.out.println(j);
System.out.println(k);
System.out.println(r);
}
public static void main(String arg[]) {
      SecondClass obj = new SecondClass();
      obj.method();
}
```

# Packages and Access Specifiers

·if SecondClass.java depends on FirstClass.java, so we need to set the classpath (cp)

**javac -d . -cp . mypackage2\SecondClass.java**

-cp . tells Java to look in the current directory (.) for other compiled classes

D:\JAVA Programs\
- mypackage1\
  - FirstClass.class
- mypackage2\
  - SecondClass.class

**Run:**

**java mypackage2.SecondClass**

# Packages and Access Specifiers

·if SecondClass.java depends on FirstClass.java, so we need to set the classpath (cp)

**javac -d . -cp . mypackage2\SecondClass.java**

-cp . tells Java to look in the current directory (.) for other compiled classes

D:\JAVA Programs\
- mypackage1\
  - FirstClass.class
- mypackage2\
  - SecondClass.class

**Run:**

**java mypackage2.SecondClass**

# Packages and Access Specifiers

.

mypackage2\SecondClass.java:7: error: k has private access in FirstClass
System.out.println(k);

mypackage2\SecondClass.java:8: error: r is not public in FirstClass; cannot be accessed from outside package
System.out.println(r);

# THANK YOU