SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

19CS204 OBJECT ORIENTED PROGRAMMING

I YEAR /II SEMESTER

Unit 1- INTRODUCTION TO OOP

Access Specifiers

Access Specifiers

- In Java, access modifiers are used to set the accessibility (visibility) of classes, interfaces, variables, methods, constructors, data members, and the setter method.

- These Specifiers determine whether a field or method in a class, can be used or invoked by another method in another class or sub-class.

- Access Specifiers can be used to restrict access. Access Specifiers are an integral part of object-oriented programming.

In java we have four Access Specifiers and they are listed below.

1. public - declarations are visible everywhere
2. private - declarations are visible within the class only
3. protected - declarations are visible within the package or all subclasses
4. default (no specifier) - declarations are visible only within the package (package private)

public

- Public Specifiers achieves the highest level of accessibility.
- Classes, methods, and fields declared as public can be accessed from any class in the Java program, whether these classes are in the same package or in another package.
- The public access modifier has no scope restriction.

public

Example

```
class student
{
  int age;
  String name;
  int rollno;
public void display()
{
  System.out.println("Age is "+age);
  System.out.println("Name is "+name);
  System.out.println("rollno is "+rollno);
}
}
```

```
public class mainstudent
{
  public static void main(String[] args)
  {
  student s=new student();
  s.age=25;
  s.name="Ryan";
  s.rollno=001;
  s.display();
  }
}
```

private

- Private Specifiers achieves the lowest level of accessibility.
- private methods and fields can only be accessed within the same class to which the methods and fields belong.
- private methods and fields are not visible within subclasses and are not inherited by subclasses.
- So, the private access specifier is opposite to the public access specifier.
- Using Private Specifier we can achieve encapsulation and hide data from the outside world.

private

Some points to be noted regarding the Private Access Modifier.

- Private access modifier cannot be used for classes and interfaces.

- The scope of private entities (methods and variables) is limited to the class in which they are declared.

- A class with a private constructor cannot create an object of the class from any other place like the main method.

private

```
class student
{
  private int age;
  String name;
  int rollno;
private void display()
{
  System.out.println("Age is "+age);
  System.out.println("Name is "+name);
  System.out.println("rollno is "+rollno);
}
}
```

```
public class mainstudent
{
  public static void main(String[] args)
  {
  student s=new student();
  s.age=25;
  s.name="Ryan";
  s.rollno=001;
  s.display();
  }
}
```

protected

- Methods and fields declared as protected can only be accessed by the subclasses in other package or any class within the package of the protected members' class.

- The protected access specifier cannot be applied to class and interfaces.

- The protected access modifier is accessible within package and outside the package but through inheritance only.

protected

```
class student
{
  private int age;
  String name;
  int rollno;
protected void display()
{
  System.out.println("Age is "+age);
  System.out.println("Name is "+name);
  System.out.println("rollno is "+rollno);
}
}
class student1 extends student
{
}
```

```
public class mainstudent
{
  public static void main(String[] args)
  {
  student1 s=new student1();
  s.age=25;
  s.name="Ryan";
  s.rollno=001;
  s.display();
  }
}
```

default

- When you don't set access specifier for the element, it will follow the default accessibility level.

- There is no default specifier keyword.

- Classes, variables, and methods can be default accessed.

- Using default specifier we can access class, method, or field which belongs to same package, but not from outside this package.

default

```
class student
{
  int age;
  String name;
  int rollno;
void display()
{
  System.out.println("Age is "+age);
  System.out.println("Name is "+name);
  System.out.println("rollno is "+rollno);
}
}
```

```
public class mainstudent
{
  public static void main(String[] args)
  {
  student s=new student();
  s.age=25;
  s.name="Ryan";
  s.rollno=001;
  s.display();
  }
}
```

THANK YOU