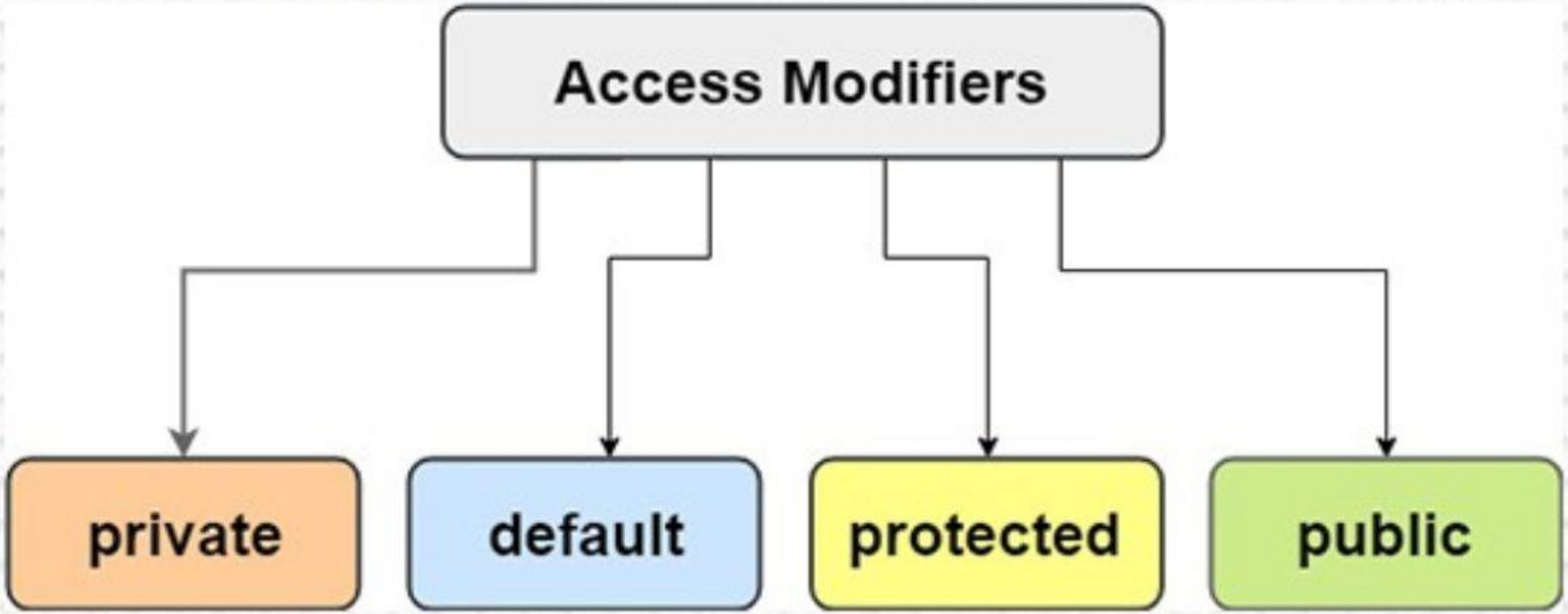
ACCESS MODIFIERS

in JAVA programming



IN THE NAME OF ALLAH

بسم الله الرحمن الرحيم

THE GRACIOUS, THE MERCIFUL.

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

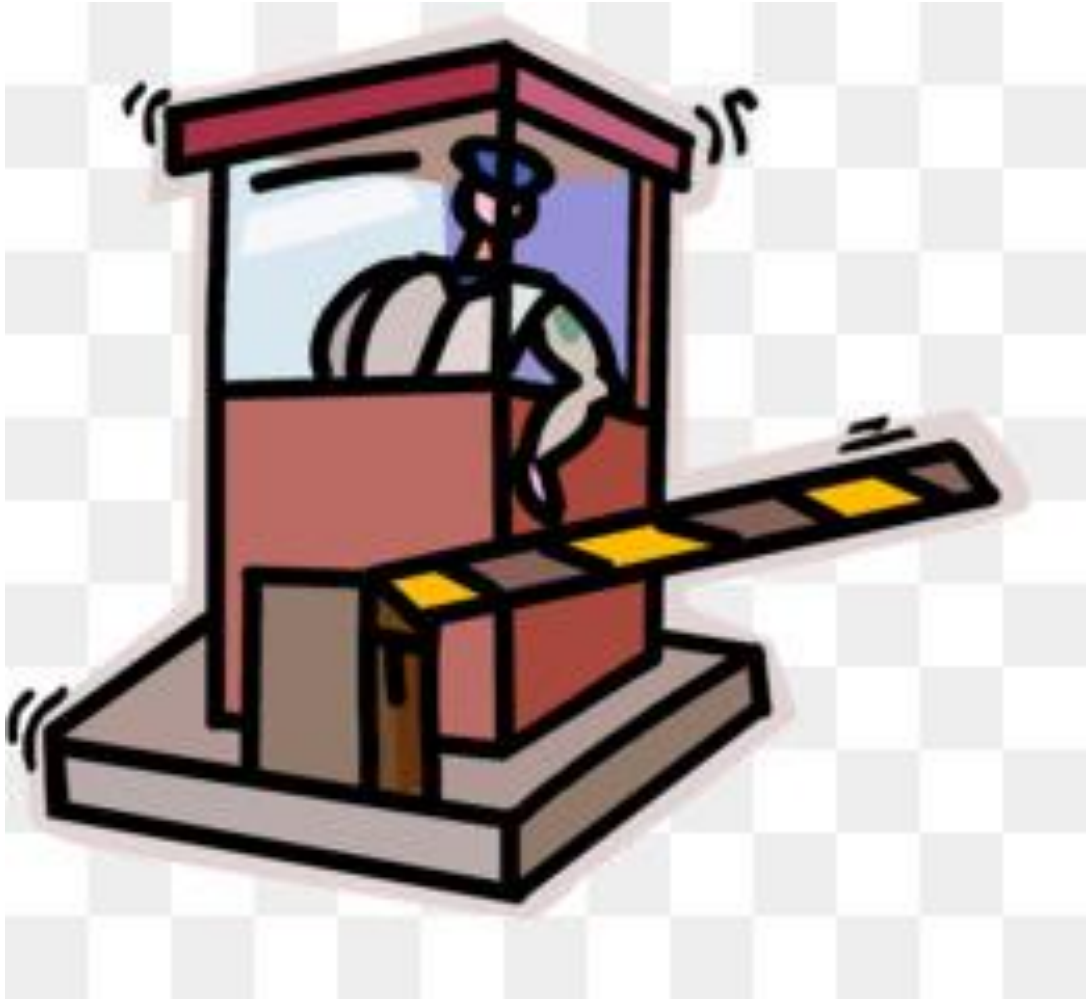
Instructor: Sir A.Rehman Ali Brohi

LECTURE: 4
Access-Modifiers

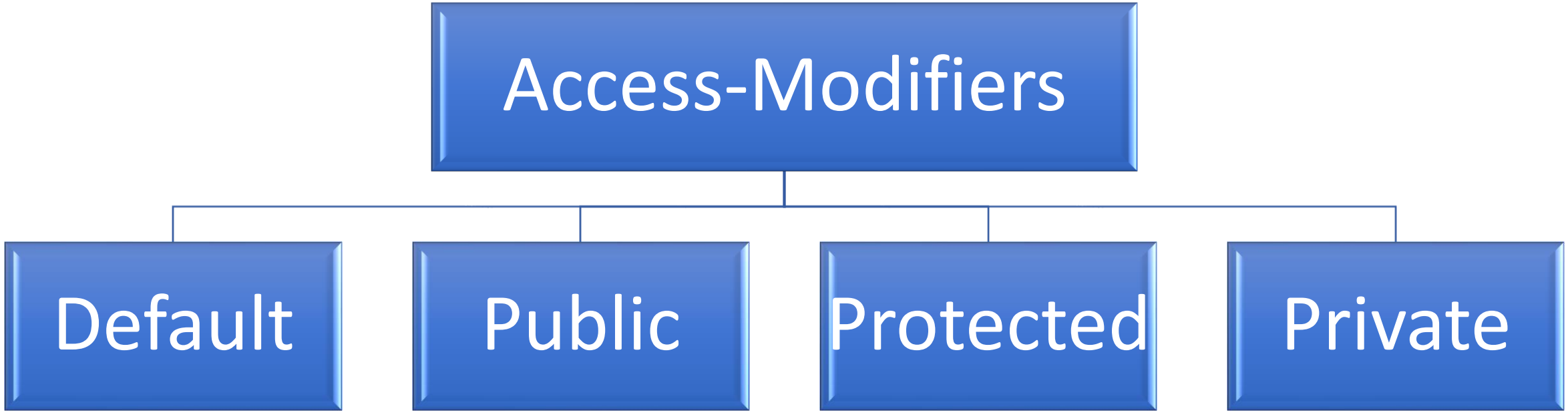
What is Access-Modifiers in java?

- They are used to control access mechanism.
- Access Modifiers in java helps to restrict the scope of:
  - Class
  - Constructor
  - Method
  - Data Member
- They also provide information about their functionality to JVM

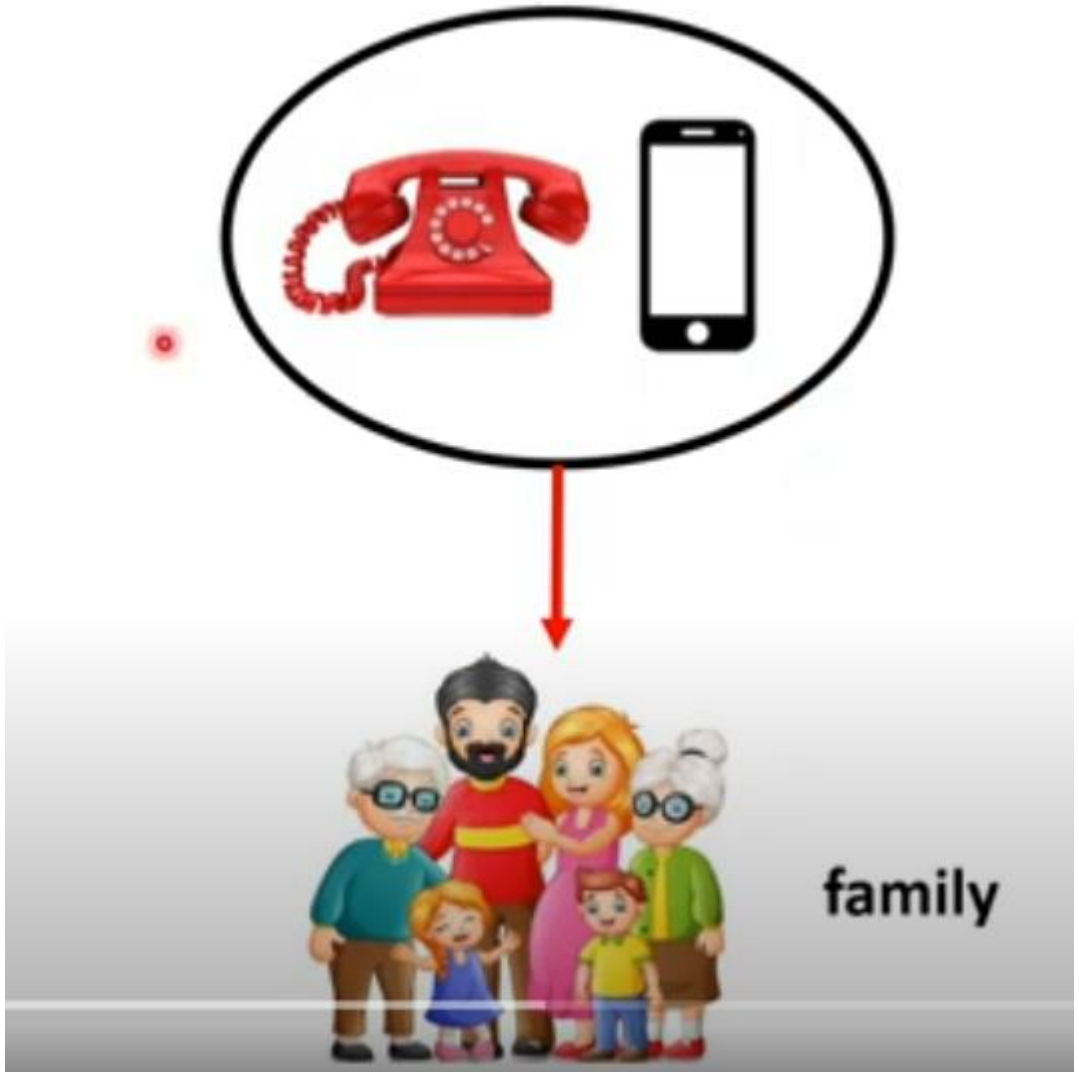
In Simple Access-Modifiers is just like a Gatekeeper



Types of Access-Modifiers



DEFAULT



PUBLIC

PRIVATE

PROTECTED

family

Known person

	Default	Private	Protected	Public
Same class	Yes	Yes	Yes	Yes
Same package sub class	Yes	No	Yes	Yes
Same package Non-sub class	Yes	No	Yes	Yes
Different package sub class	No	No	Yes	Yes
Different package Non-sub class	No	No	No	Yes

Default (called package-private).

- When no access modifier is specified for a class , method or data member – I is said to be having the **default** access modifier by default.
- The data members, class or methods having default access modifier are accessible **only within the same package**.

OR

- We cannot access data members, class or methods **from one package to the other**

Default (Example)

```
Package one;

class Codeera{

void msg()
  {
    System.out.println(" Welcome to Codeera");
  }

}
```

```
Package two;
Import one.*;

class Codeera1{

public static void main(String args[]){

Codeera c= new Codeera();
c.msg();
}
}
```

Compile time error ← **OUTPUT**

Public

- ***Public*** Keyword is used to specify the public access modifier.
- Classes, methods or data members which are declared as public are **accessible from every where** in the program.
- There is no restriction on the scope of a public data members it has the widest scope

Public (Example)

```
Package one;

Public class Codeera{

Public void msg()
  {
    System.out.println(" Welcome to Codeera");
  }

}
```

```
Package two;
Import one.*;

class Codeera1{

Public static void main(String args[]){

Codeera c= new Codeera();
c.msg();
}
}
```

Welcome to Codeera ← **OUTPUT**

Protected

- ***Protected*** Keyword is used to specify the protected access modifier.
- The methods or data members declared as protected are **accessible within same package or sub classes in different package.**
- **protected** applies only when inheritance is involved.

PROTECTED → "only visible within the enclosing class and any subclasses".

Protected (Example)

```
Package one;

class Codeera{

Protected void msg()
  {
    System.out.println(" Welcome to Codeera");
  }

}
```

```
Package two;
Import one.*;

class Codeera1 extends Codeera{

Public static void main(String args[]){

Codeera1 c= new Codeera1();
c.msg();
}
}
```

Welcome to Codeera ← OUTPUT

Private

- ***Private*** Keyword is used to specify the private access modifier.
- The methods or data members declared as private are accessible only **within the class** in which they are declared.
- Any other **class of same package will not be able to access** these members.
- Top level Classes or interface can not be declared as private .

Private (Example)

```
Package one;

class Codeera {

private void msg()
  {
    System.out.println(" Welcome to Codeera");
  }

class Codeera1 {

public static void main(String args[]) {

   Codeera c= new Codeera();
   c.msg();
}
}
```

Compile time error

msg() has private access in Codeera

Assaignment :

1. Write their accessible like above table 3 times.

2. Write some of their example Programs