**1. What are the four access modifiers available in Java and what is their significance in**

**terms of class, method, and variable accessibility**

Access modifiers are defining the visibility or accessibility of any classes, methods, variables and constructors in java. They determine how other classes and objects can interact with members of a class.

There are four access modifiers:

i) Default

ii) Public

iii) Private

iv) Protected

**DEFAULT**

When we don’t use any keyword explicitly, Java will set a **default** access to a given class, method and property. The default access modifier is also called as package-private, which means that all members are visible within the same package can’t access any members in different package.

Default access modifier with data members:

Class Accessmodifier { default type class

Int age; default instance variable

String name;

//Method inside the class

Static void defaultMethod() { default type method

//Method body

}

}

**PUBLIC**

If we add the **public** keyword to a class, method, or property, then we’re making it available to the whole java project, which is all other classes in all packages will be able to use it. This is the least restrictive access modifier.

**Public** access modifier with data members:

Public Class Accessmodifier { public type class

public int age; public instance variable

public String name;

//Method inside the class

Public static void defaultMethod() { public method

//Method body}

}

**PRIVATE**

Any method, property, or constructor with the private keyword is accessible from the same class only. This is the most restrictive access modifier, and is core to the concept of encapsulation. All data will be hidden from the outside world.

**private** access modifier with data members:

private Class Accessmodifier { private type class

private int age; private instance variable

private String name;

//Method inside the class

private static void defaultMethod() { private method

//Method body

}

}

**PROTECTED**

Between public and private access levels, there’s the protected access modifier.

If we declare a method, property, or constructor with the protected keyword, we can access the member from the same package as well as from all subclasses of its class, even if they lie in other packages.

**protected** access modifier with data members:

**protected** Class Accessmodifier { **protected** type class

**protected** int age; **protected** instance var

**protected** String name;

//Method inside the class

**protected** static void defaultMethod() { **protected** method

//Method body

}

}