To understand the work of **@Primary** annotation let’s understand a scenario.

Suppose, **Student** class has a dependency of **Pen** interface and **Pen** interface has two implementation classes **BlackPen** and **BluePen**.

**@Component**

class BlackPen implements Pen { }

**@Component**

class BluePen implements Pen { }

class Student {

private Pen pen;

**@Autowired**

public void setPen(Pen pen) {

this.pen = pen;

}

}

Now if we use **@Autowired** annotation over the setter of **Pen** dependency in **Student** class then spring will got confuse which implantation of **Pen** it should inject.

Now one solution of the above problem can be use of **@Qualifier** annotation.

class Student {

private Pen pen;

**@Autowired**

**@Qualifier(“bluePen”)**

public void setPen(Pen pen) {

this.pen = pen;

}

}

But how **@Primary** will remove this problem ?

So, we just have to use **@Primary** annotation with that implementation of **Pen** interface which you want to be first preference out of multiple implementations.

For ex : if we use **@Primary** annotation with **BlackPen** class

**@Component**

**@Primary**

class BlackPen implements Pen { }

now even if we are not telling spring that which implementation you should use still spring will run our code and use the Primary implementation of **Pen** interface i.e. **BlackPen.**

For more project details visit : **#12-@Primary**