

- **Remove XML or Without XML Configuration : -**

- 1) @Configuration
- 2) @ComponentScan
- 3) @Bean

Previously when we was creating application we was using XML file that was provide data and object to IOC container but now in this example we are not using XML anymore we can do it with above notations.

```
n/src/main/java/com/spring/javaconfig/Student.java - Eclipse IDE
ch Project Run Window Help

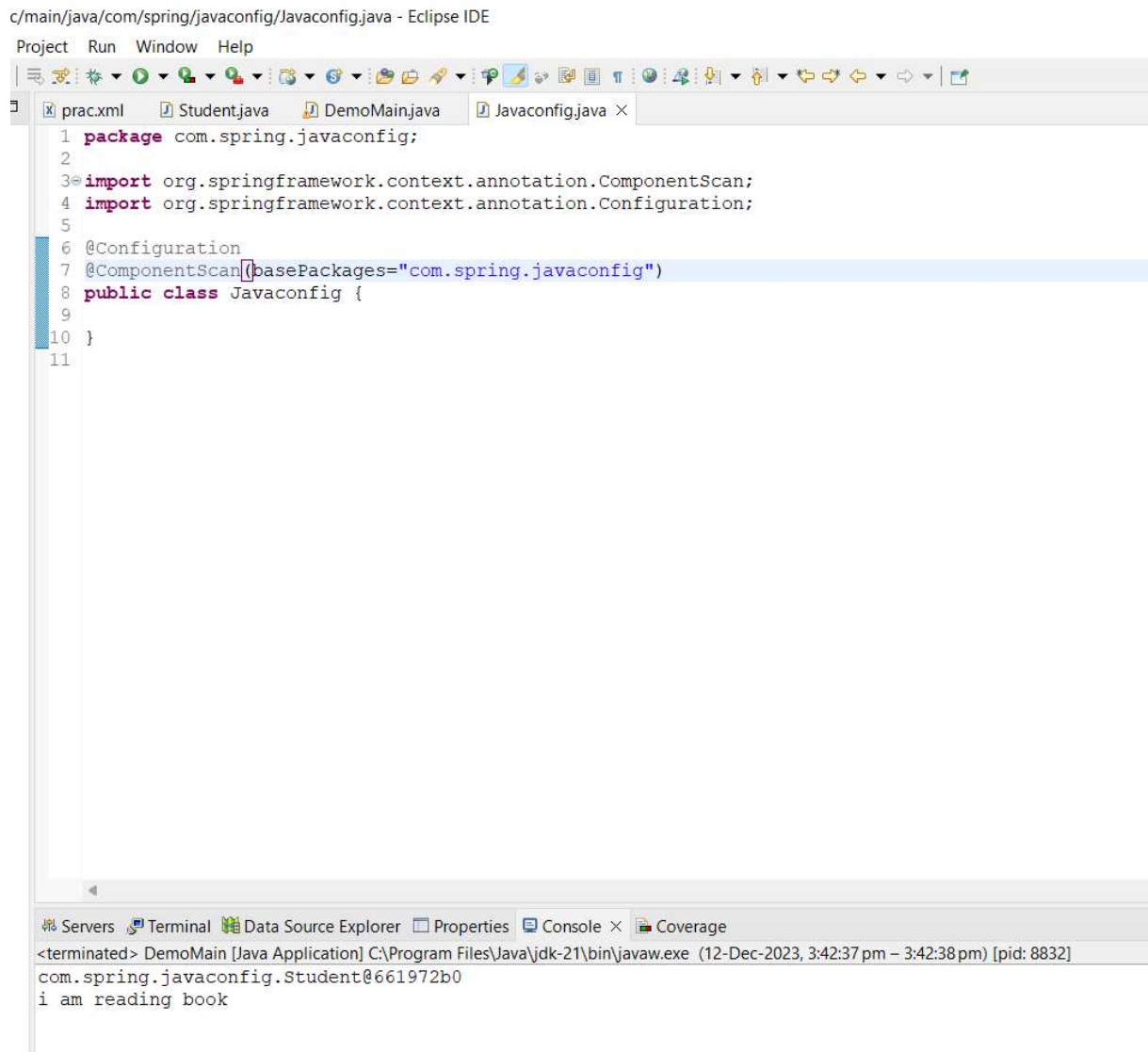
1 package com.spring.javaconfig;
2
3 import org.springframework.stereotype.Component;
4
5 @Component("ob1")
6 public class Student {
7     public void show()
8     {
9         System.out.println("i am reading book");
10    }
11 }
12
13

n

Servers Terminal Data Source Explorer Properties Console x Coverage
<terminated> DemoMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (12-Dec-2023, 3:42:37 pm - 3:42:38 pm) [pid: 8832]
com.spring.javaconfig.Student@661972b0
i am reading book
```

```
/main/java/com/spring/javaconfig/DemoMain.java - Eclipse IDE
Project Run Window Help

1 package com.spring.javaconfig;
2 import org.springframework.context.ApplicationContext;
3 import org.springframework.context.support.ClassPathXmlApplicationContext;
4 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
5 public class DemoMain {
6
7     public static void main(String[] args) {
8         ApplicationContext con=new AnnotationConfigApplicationContext(Javaconfig.class);
9         Student stu=con.getBean("ob1", Student.class);
10        System.out.println(stu);
11        stu.show();
12    }
13 }
```



The screenshot shows the Eclipse IDE interface. The top toolbar includes icons for Project, Run, Window, and Help. The editor window displays the file `Javaconfig.java` with the following code:

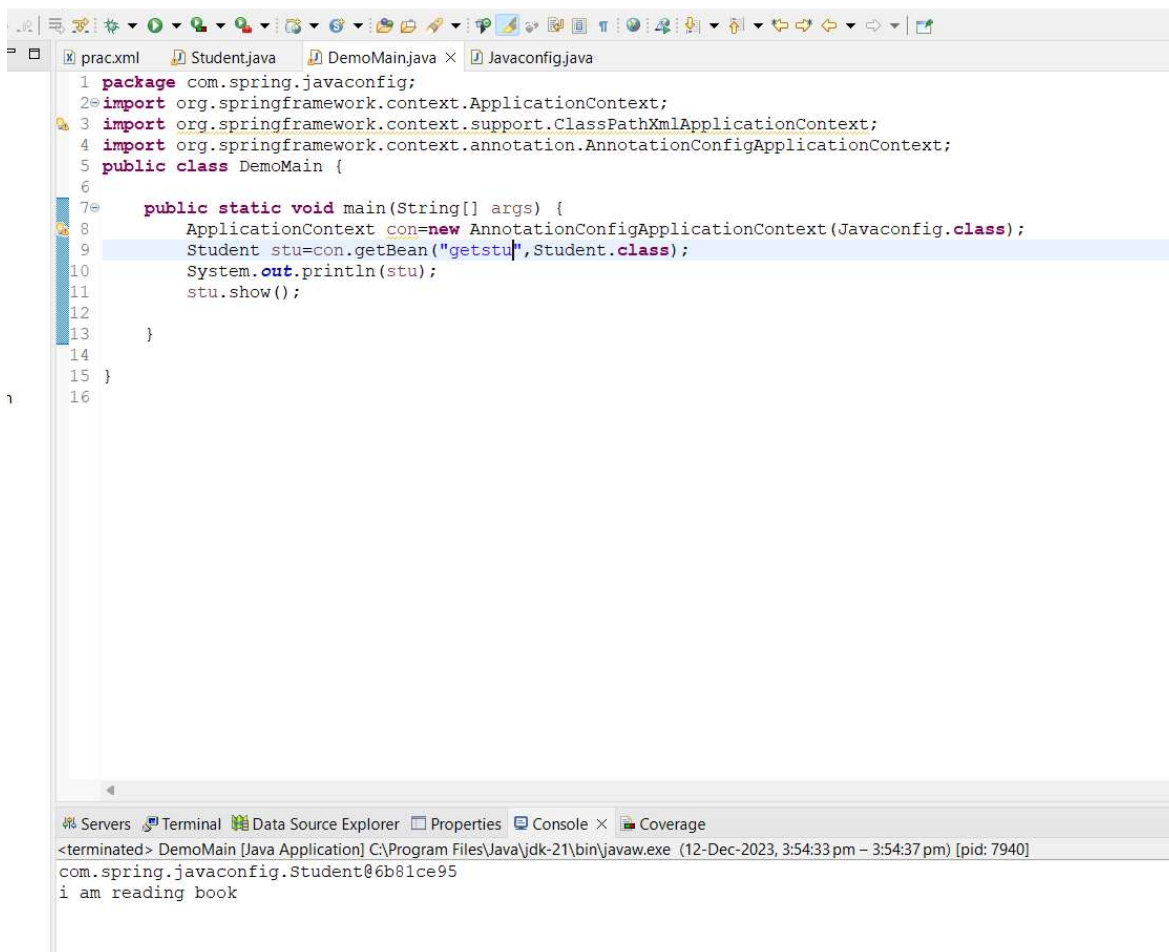
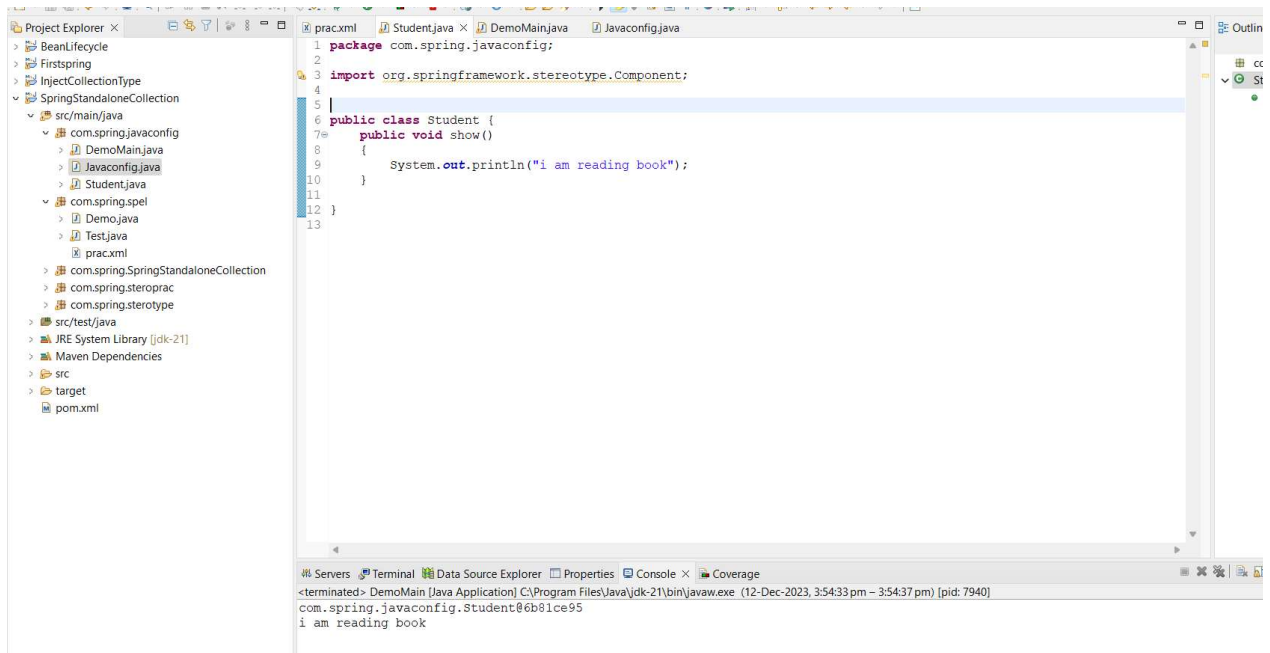
```
1 package com.spring.javaconfig;
2
3 import org.springframework.context.annotation.ComponentScan;
4 import org.springframework.context.annotation.Configuration;
5
6 @Configuration
7 @ComponentScan(basePackages="com.spring.javaconfig")
8 public class Javaconfig {
9
10 }
11
```

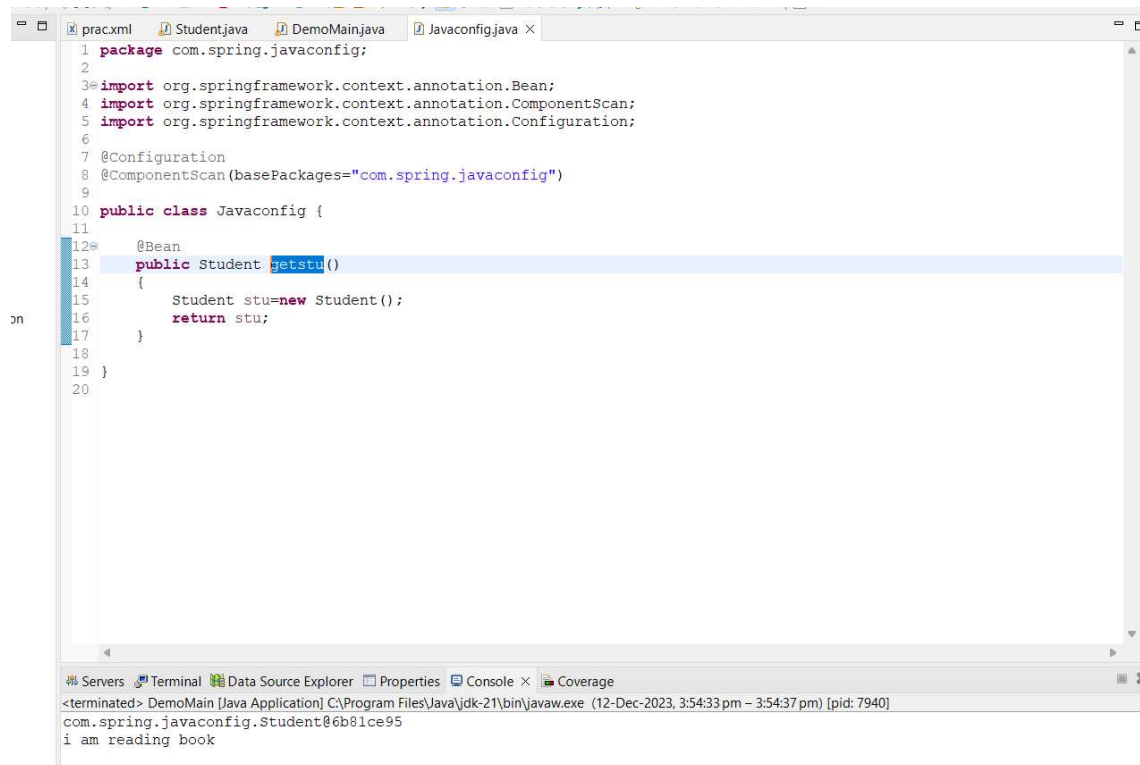
The bottom panel shows the Console view with the following output:

```
<terminated> DemoMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (12-Dec-2023, 3:42:37 pm - 3:42:38 pm) [pid: 8832]
com.spring.javaconfig.Student@661972b0
i am reading book
```

@Bean : it is written above method that is returning object to indicate that object is given to IOC container.with the help of bean notation object will go to the IOC container.

Now u dont need to specify @component annotation

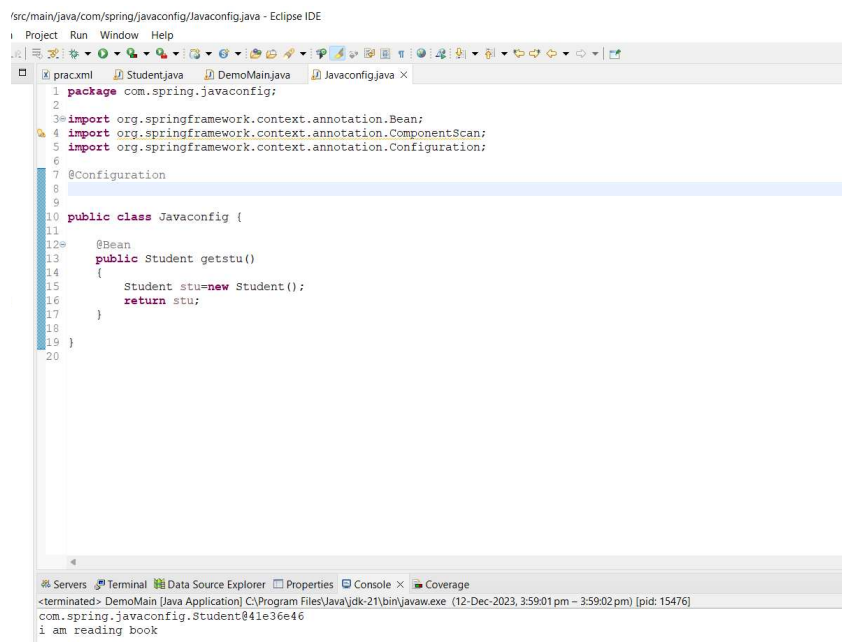




```
1 package com.spring.javaconfig;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.ComponentScan;
5 import org.springframework.context.annotation.Configuration;
6
7 @Configuration
8 @ComponentScan(basePackages="com.spring.javaconfig")
9
10 public class Javaconfig {
11
12     @Bean
13     public Student getstu()
14     {
15         Student stu=new Student();
16         return stu;
17     }
18
19 }
20
```

com.spring.javaconfig.Student@6b81ce95
i am reading book

Even you can run it without @ComponentScan Annotation



```
1 package com.spring.javaconfig;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.ComponentScan;
5 import org.springframework.context.annotation.Configuration;
6
7 @Configuration
8
9
10 public class Javaconfig {
11
12     @Bean
13     public Student getstu()
14     {
15         Student stu=new Student();
16         return stu;
17     }
18
19 }
20
```

com.spring.javaconfig.Student@41e36e46
i am reading book

INjecting dependencies :-

main/java/com/spring/javaconfig/Student.java - Eclipse IDE

```
1 package com.spring.javaconfig;
2
3 import org.springframework.stereotype.Component;
4
5
6 public class Student {
7     private Samosa samo;
8
9
10    public Samosa getSamo() {
11        return samo;
12    }
13
14
15    public void setSamo(Samosa samo) {
16        this.samo = samo;
17    }
18
19
20
21    public Student(Samosa samo) {
22        super();
23        this.samo = samo;
24    }
25
26
27    public void show()
28    {
29        this.samo.display();
30        System.out.println("i am reading book");
31    }
32
33 }
34
```

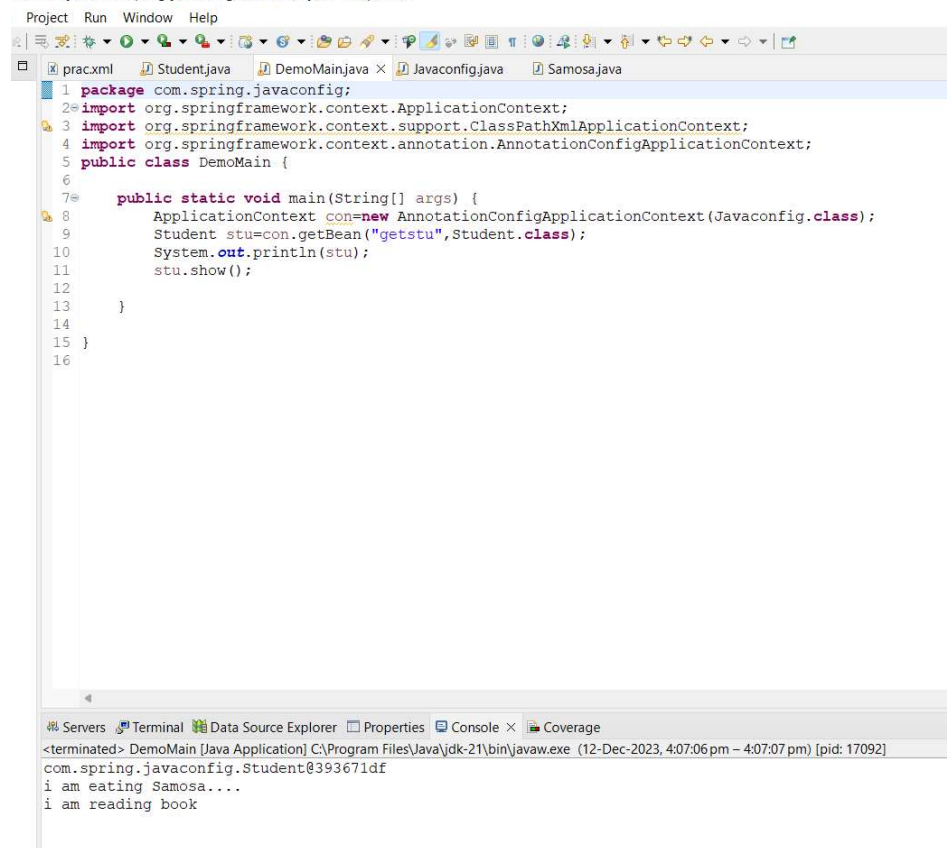
Servers | Terminal | Data Source Explorer | Properties | Console | Coverage

```
<terminated> DemoMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (12-Dec-2023, 4:07:06 pm - 4:07:07 pm) [pid: 17092]
com.spring.javaconfig.Student@393671df
i am eating Samosa....
i am reading book
```

n/src/main/java/com/spring/javaconfig/Samosa.java - Eclipse IDE

```
1 package com.spring.javaconfig;
2
3 public class Samosa {
4     public void display()
5     {
6         System.out.println("i am eating Samosa....");
7     }
8
9 }
10
```

src/main/java/com/spring/javaconfig/DemoMain.java - Eclipse IDE



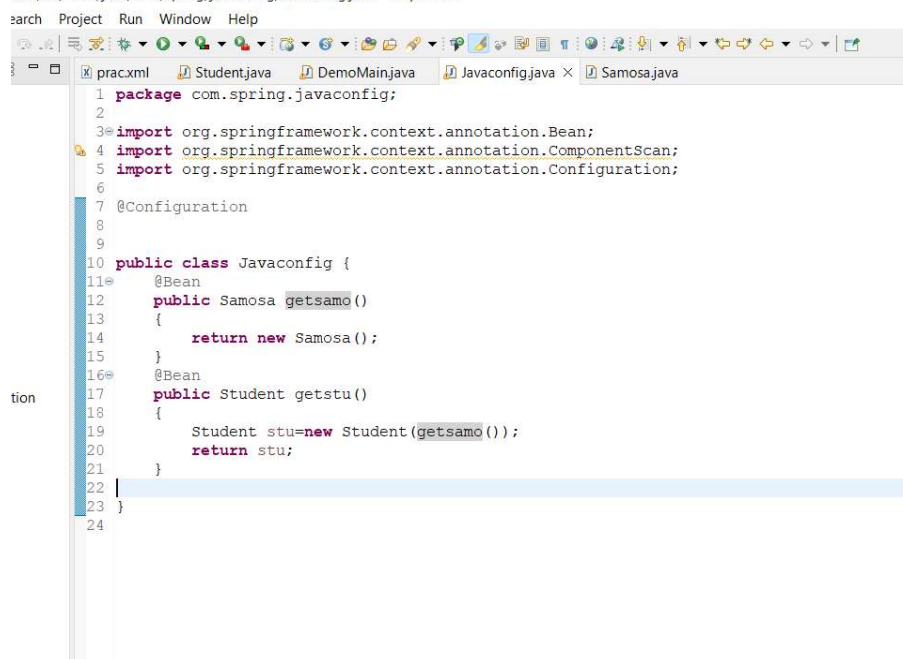
The screenshot shows the Eclipse IDE interface. The top toolbar includes icons for Project, Run, Window, and Help. The editor window displays the file DemoMain.java with the following code:

```
1 package com.spring.javaconfig;
2 import org.springframework.context.ApplicationContext;
3 import org.springframework.context.support.ClassPathXmlApplicationContext;
4 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
5 public class DemoMain {
6
7     public static void main(String[] args) {
8         ApplicationContext con=new AnnotationConfigApplicationContext(Javaconfig.class);
9         Student stu=con.getBean("getstu",Student.class);
10        System.out.println(stu);
11        stu.show();
12    }
13 }
14
15 }
16
```

The bottom console window shows the output of the application:

```
<terminated> DemoMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (12-Dec-2023, 4:07:06 pm - 4:07:07 pm) [pid: 17092]
com.spring.javaconfig.Student@393671df
i am eating Samosa....
i am reading book
```

tion/src/main/java/com/spring/javaconfig/Javaconfig.java - Eclipse IDE



The screenshot shows the Eclipse IDE interface. The top toolbar includes icons for Project, Run, Window, and Help. The editor window displays the file Javaconfig.java with the following code:

```
1 package com.spring.javaconfig;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.ComponentScan;
5 import org.springframework.context.annotation.Configuration;
6
7 @Configuration
8
9
10 public class Javaconfig {
11     @Bean
12     public Samosa getsamo()
13     {
14         return new Samosa();
15     }
16     @Bean
17     public Student getstu()
18     {
19         Student stu=new Student(getsamo());
20         return stu;
21     }
22 }
23
24
```

As we have saw that we are specifying methodname of Bean in getbeans method to inject dependency but we can specify name by ourself also just we have to add parameter in @Bean annotation

```
@Bean(name={"om","shubh","anu"})
```