JavaBased:

JavaBased program means in this program we can not used .xml file we used java class in place of .xml file for bean creation of the pojo class.

@Configuration

This annotation is used on classes that define beans. @Configuration is an analog for an XML configuration file – it is configured using Java classes. A Java class annotated with @Configuration is a configuration by itself and will have methods to instantiate and configure the dependencies.

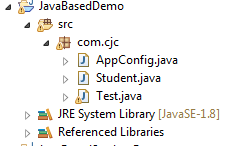
@Autowired

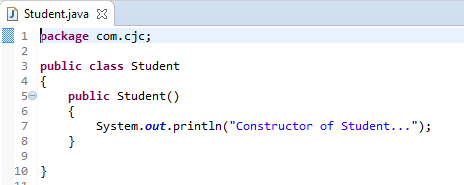
This annotation is applied to fields, setter methods, and constructors. The @Autowired annotation injects object dependency implicitly.

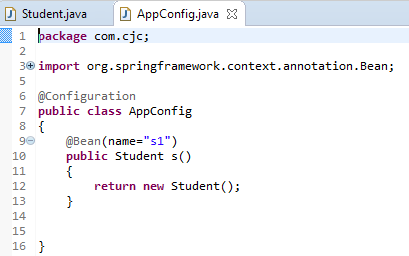
When you use @Autowired on fields and pass the values for the fields using the property name, Spring will automatically assign the fields with the passed values.

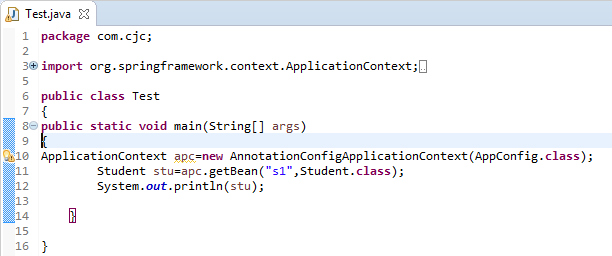
### @Bean

This annotation is used at the method level. The @Bean annotation works with @Configuration to create Spring beans. As mentioned earlier, @Configuration  will have methods to instantiate and configure dependencies. Such methods will be annotated with @Bean. The method annotated with this annotation works as the bean ID, and it creates and returns the actual bean.

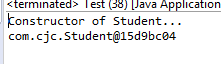




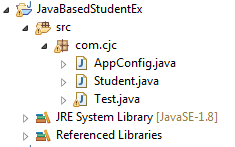


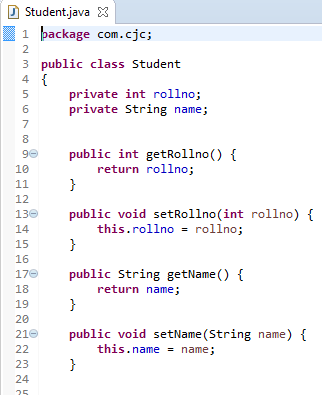


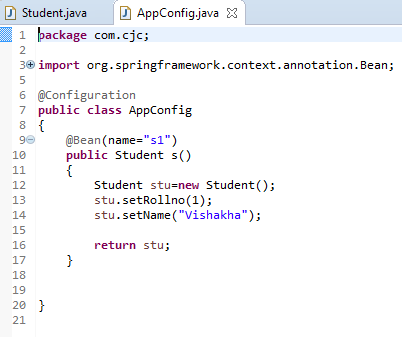
O/P:

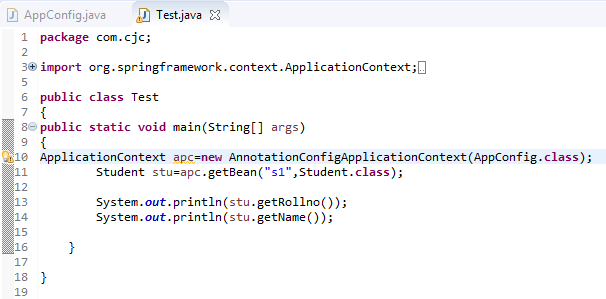


Task 2:



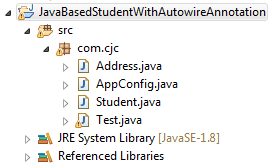


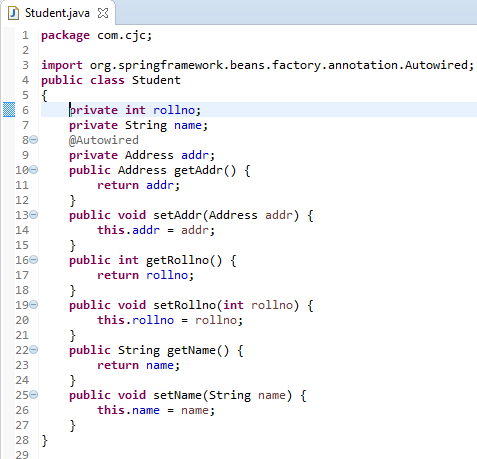


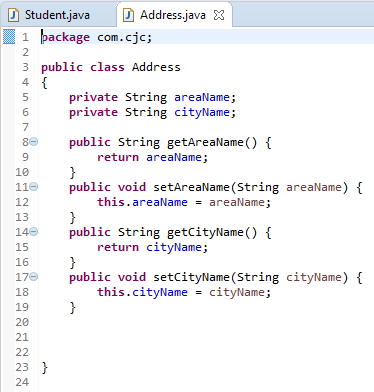


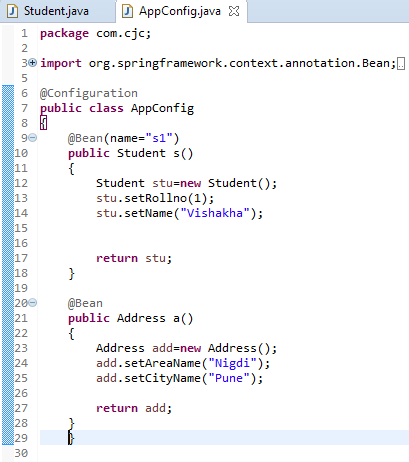
O/P:

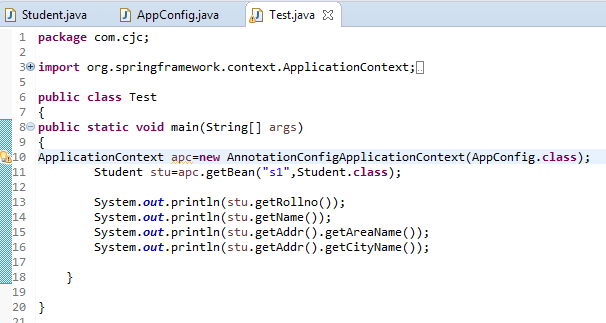








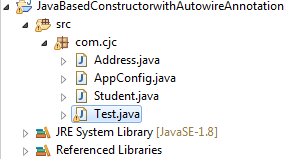


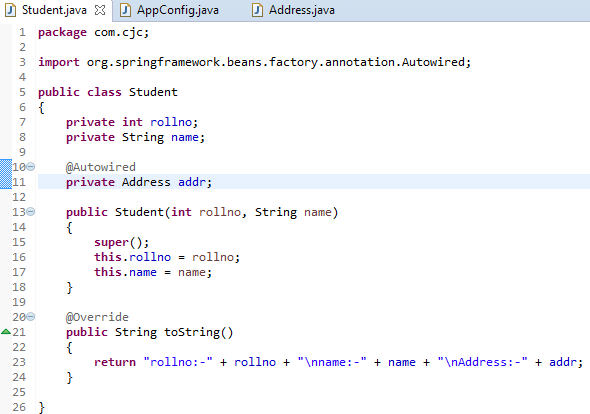


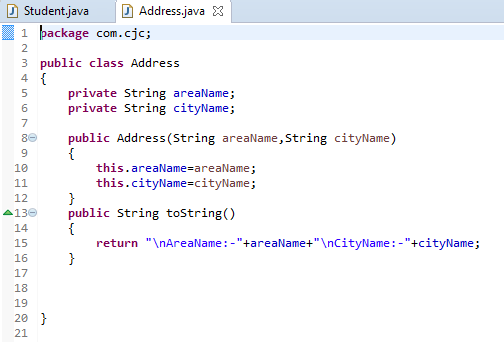
O/P:

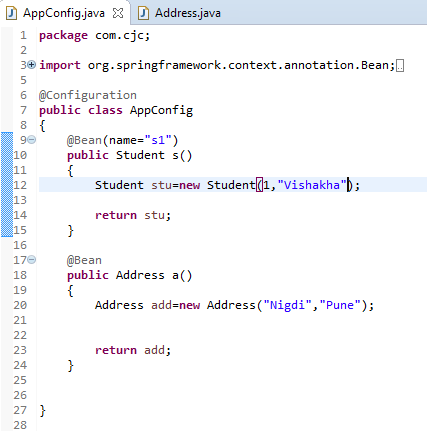


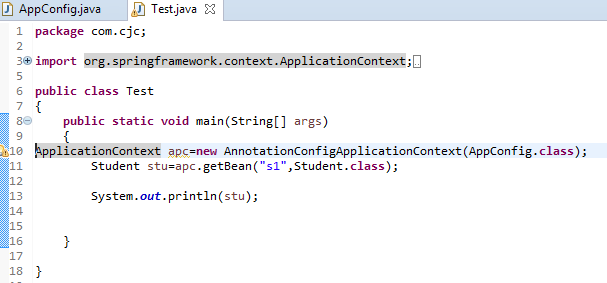
Task4:



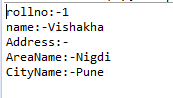








O/P:



SpringMVC :

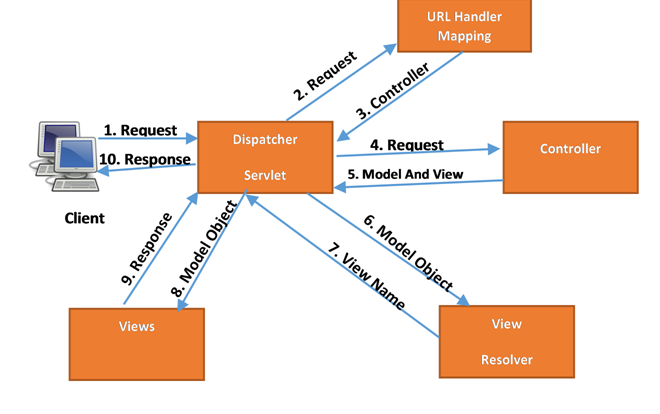
The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

This module of spring framework help us to use MVC in spring.MVC pattern allows us to separate the different aspect of the application program i.e. input logic, business logic, and UI logic and it also provide the loose coupling between these aspects.

M: Model :-It will bind the all data of application i.e. POJO

V: View (Html, Jsp):-It generates the HTML output with the help of model that the client’s browser can interprete.

C: Controller(Servlet):-It processes the user request and build model passes it to view for processing.



* First request will be recieved by DispatcherServlet. DispatcherServlet is Front Controller of Spring MVC.It is responsible to manage the flow of the Spring MVC application.
* DispatcherServlet will come to know about corresponding Controller class with the help of HandlerMapping.
* Now Controller will process the request and returns ModelAndView object to DispatcherServlet.
* Now with the help of ViewResolver, DispatcherServlet will get actual view page.
* At last, DispatcherServlet will pass that object to View page and generates response i.e. View.

All the above mentioned components ie. HandlerMapping, Controller and ViewResolver are parts of WebApplicationContext which is an extension of the plain ApplicationContext with some extra features necessary for web applications.

Dispatcher Servlet: DispatcherServlet delegates the request to the controllers to execute the functionality specific to it

DispatcherServlet is a single servlet that manages the entire request-handling process. When a request is sent to the DispatcherServlet it delegates the job by invoking the appropriate controllers to process the request. Like any other servlet the DispatcherServlet need to be configured in the web.xml

Controller: The **@Controller** annotation indicates that a particular class serves the role of a controller.

The **@RequestMapping** annotation is used to map a URL to either an entire class or a particular handler method.

@**RequestMapping**(value=”/url”, method= RequestMethod.GET)

The **value** attribute indicates the URL to which the handler method is mapped and the **method** attribute defines the service method to handle HTTP GET request.

JSP Pages: Spring MVC supports many types of views for different presentation technologies. These include - JSPs, HTML, PDF, Excel worksheets, XML, Velocity templates, XSLT, JSON, Atom and RSS feeds, JasperReports etc.

ModelAndView : The Controller process the request by calling the appropriate service methods and returns aModeAndView object to the DispatcherServlet. The ModeAndView object contains the model data and the view name.

Spring Mvc First Program with string:

