```
webapp's
                                      Controller (B.L)
                                                                    Model
  client(browser)
                     request
                                          #2
                                                          #3
                                       Servlet
                                                                    JDBC
                                       #5a
                                               #5b
                    response
                                          JSP
                                View
                                                     autocompile)
                                     dyanmic presentation
                                      browser
                                  req (h) (h) read student object data and print the student
                                                    details in table format
                                    Servlet
                                              resp(student)
                                      JDBC <=====> DB(Student Data)
How JSP Programs would get executed?
 set classpath with 2 jars
                                                          javax.servlet.Servlet(I)
   a. jsp-api.jar
                                            extends
   b. jasper.jar

    ↑ implements

                                                                                                     init(ServletConfig config)
                   jspInit()
                                                                                                     service(ServletRequest,ServletResponse)
                                                                            GenericServlet(AC)
                              javax.servlet.jsp.JspPage(I)
                   jspDestroy()
                                                                                                     destroy()
                                                                                   getServletConfig()

♠ extends

                                                                             HttpServlet(AC)
                                                                                                     getServletInfo()
         _jspService(req,resp)
                             javax.servlet.jsp.HttpJspPage(I)

☆ implements

                                                                          extends
            _jspInit()
            _jspDestroy() org.apache.jasper.runtime.HttpJspBase(AC)
                                     translated

    Translated Servlet loading

       index.jsp
                                 index_jsp.java
                                                                    Translated Serlvet Instantiation
                                                                 4. Translated Servlet requestProcessing → _jspService(req,resp)

    Translated Servlet DeInstantiation ⇒ jspDestroy()

  location of translated servlet
 C:\Tomcat 9.0\work\Catalina\localhost\SecondApp\org\apache\jsp\index_jsp.java
Note: _symbol represents that this method is generated automatically by JSP Engine and we cannot write explicitly
Life Cycle of JSP
                     Translation of JSP to Servlet
     First
    Request
                                                            Even before the request comes, if we want the translation phase to complete then we
                                                            need to precompile the jsp page as shown below.
                                                              http://localhost:9999/SecondApp/index.jsp?jsp_precompile=true
                     Compilation Phase(.java to .class)
                                                           Advantage
                                                            a. All the request will be processed with uniform response time.
                           Class Loading
       Jasper
                            Instantiation
                               रु
     catalina
                            jspInit()
    For 2nd
                        _jspService(req,resp)
                                                            response
    request
                            jspDestroy()
  JSP Elements
                                                  JspPage
                                                                             JSPElements
                   Template Text
                                                                                                  ScriptingElements
                             Directives
                                                           Actions
                                 -page
-include
                                 _taglib
                                                                                          Traditional
                                                                                                                         Modern
                                                StandardActions
                                                                      CustomActions
                                                                                                                    1.ExpressionLanguage

    Scriptlet

                                              <jsp:useBean>
                                                                       <mine:mytag/>
                                                                                                                          ${x}
                                                                                            Expression
                                              <jsp:setProperty>
                                                                                            Declaration
                                              <jsp:getProperty>
                                              <jsp:include>
                                                                                            Comments
                                              <jsp:forward>
                                                                                                 a. jsp comments
                                                                                                 b. html comments
                                              <jsp:param>
                                              <jsp:fallback>
                                                                                                 c. java comments
 Template Text
    It contains plain text data and html tags
    For template text no processing is requried and it will become argument to write() method in
      _jspService(req,res) method
   index.jsp
   =======
   <h1>The server time is <%=new java.util.Date()%></h1>
                  Translated servlet
     public final class index_jsp extends .....
         _jspService(request,response){
                                                           write() => it takes only character data
                  out.write("The server time is ");
                                                           println() => it can take any type of argument
                  out.println(new java.util.Date())
  Directives
  =======

    page directive

      In the current jsp page if we want to define import statements, present jsp page characterstics
       then we need to go for page directive.
          syntax:
             <%@ page [attribute-list]%>
   a. language ='java'
       This is the default value of language attribute
   b. contentType = 'text/html'
        This is the default value of contentType attribute
        As per the page requirement we can change the values too.
  c. import = ''
       This is the only attribute which can be repeated in the following ways
            eg: <%@ page import = '' import = '' %>
                <%@ page import = 'java.io.*,java.util.Date'%>
                <%@ page import = 'java.io.*'%>
                <%@ page import = 'java.util.*'%>
        The default values of import is
             java.lang.*
             javax.servlet.*;
                                                                                                                      autoFlush= true
             javax.servlet.http.*;
                                                                                                                          52kb
             javax.servlet.jsp.*;
                                                                                                                         buffer
                                                                                                        52kb is full
    d. extends
                                                                                                         .jsp
            The default value of extends in "HttpJspBase".
    e. info
                                                                                                                                           console
          The default value of this attribute is "Jasper JSP2.3 Engine"
          This value can be obtained inside the servlet by making a call to
           method called "getServletInfo()".
   f. buffer, autoFlush
        buffer => jsp internally maintains a buffer to write the data to the console as
                                                                                                                      autoFlush=false
                  the response.
                                                                                                                          52kb
                  The default size of the buffer is 8kb.
                                                                                                                         buffer
                                                                                                        52kb is full
        autoFlush => It is a boolean attribute which is used to indicate to the container
                    whether to flush the response to client automatically or not.
                    if autoFlush is true, then container will flush the complete response
                     to the client from the buffer.
                                                                                                                                           console
                    if autoFlush is false, then container will raise an Exception when the
                                                                                                                                          Exception
                    Buffer is filled with response.
            java.io.IOException: Error: JSP Buffer overflow
```