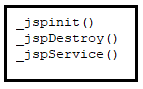
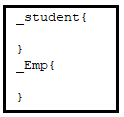
* The method name /class name which begins with “\_” is container specific method/container specific class. They have only container specific logic.

Ex:- container specific methods



Ex:- Container specific classes



**1.JSP Life cycle Methods:**

===================

1. jspInit():- It is for placing programmer choice initialization logic like creating jdbc con object.
2. \_jspInit():- It is for placing container specific initialization logic like creating EL Engine.

Note:- The jspInit() and \_JspInit() methods are invoked through servlet life cycle method init(ServletConfig cg).

1. \_jspService(-,-) :- It is for placing business processing logic . The programmer writes java code in .jsp file. That is collected and placed in \_jspService(-,-) of JES class by Jspc.

The \_jspService(-,-) of JES class contains

1. Logics to create implicit objs.
2. Code collected from scriplet and other jsp tags like expression tags.
3. Template text.
4. Exception handling.
5. Etc..

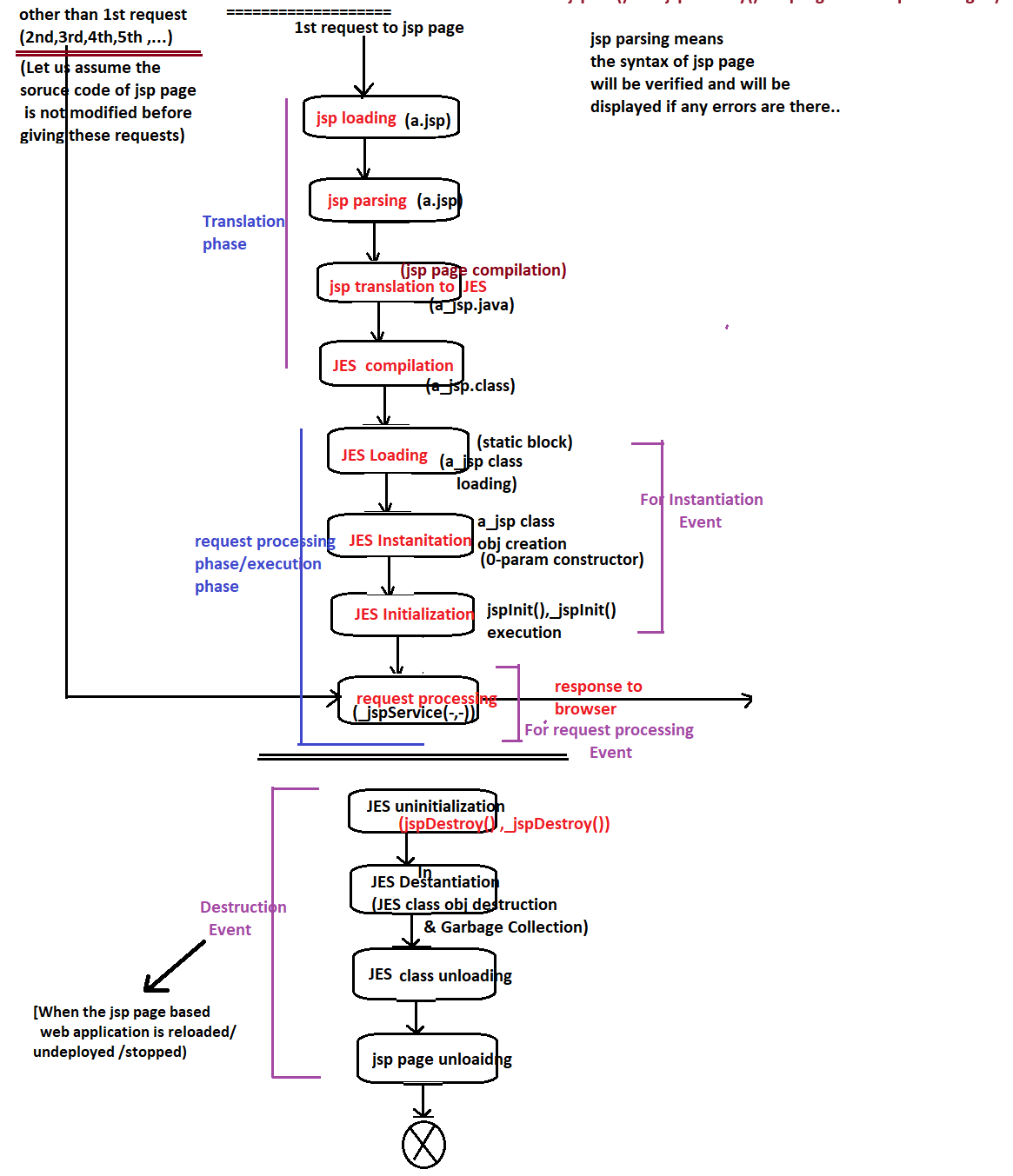
Note:-it is invoked through servlet life cycle method Service(-,-)

1. jspDestroy():- It is for placing programmer choice uninitialization logic like closing jdbc con obj.
2. \_jspDestroy():- It is for placing container specific unitialization logic like closing EL engine.

Note:- jspDestroy(),\_jspDestroy() are invoked through servlet life cycle method destroy().

2.JSP Life cycle Diagram:

==================

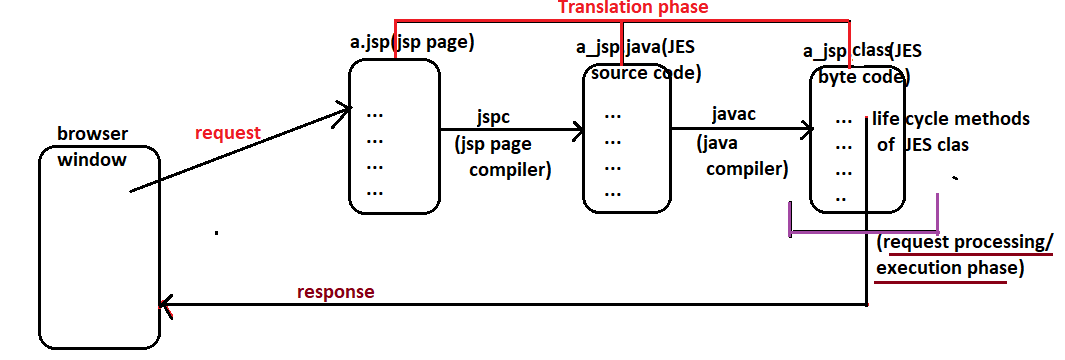


There are two phases:

1. Translation Phase.
2. Request Processing/Execution Phase.

Translation Phase:- In this phase the jsp page will be converted into equalent servlet source code and byte code.

Request Processing/Execution phase:- in this phase the byte code of JES class executes by calling its life cycle methods and generated output goes to browser as response.



The request given to jsp page directly participates in execution phase/request processing if the byte code of JES class is already available and jsp page source code is not modified before giving the request otherwise the request given to jsp page first participates in translation phase and then participates in request processing phase.

The JSP container internally maintains time stamping of JES source code and byte code .. i.e though we delete source code and byte code .. the generated source code and byte code will have old date and time if the jsp page source code is not modified.

Ex:- delete .java, .class files of JES class with out modifying .jsp page source code. Then give new requests and generated JES class .java, .class files date an d time. Surprisingly we will see old date and time.

Q) if we give request to jsp page with out modifying source and by just deleting source code file(.java file) then what happens?

A) the new .java file will not be created for next request because the container will check only for .class file availability and jsp page source code modificiation.