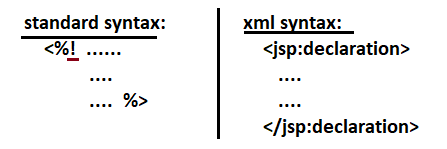
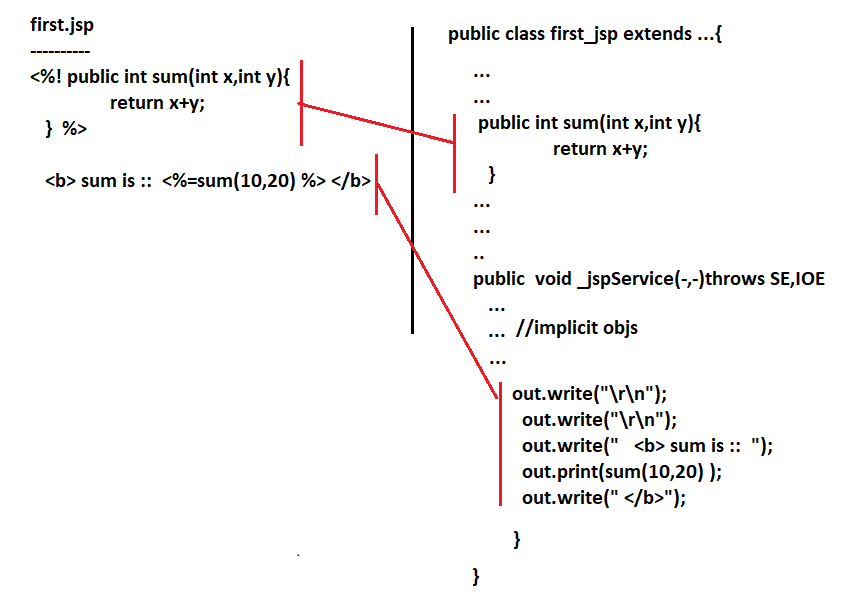
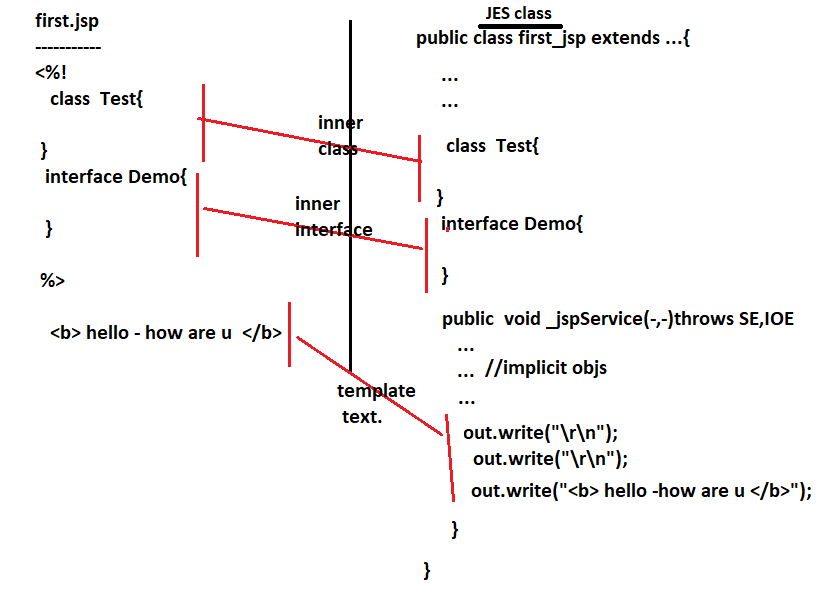
* The code which is placed in declaration tag is placed outside of \_jspService(-,-) method and in side JES class.
* The global var decl , method definitations, class definitation , interfaces definitation can be written in declaration tag.



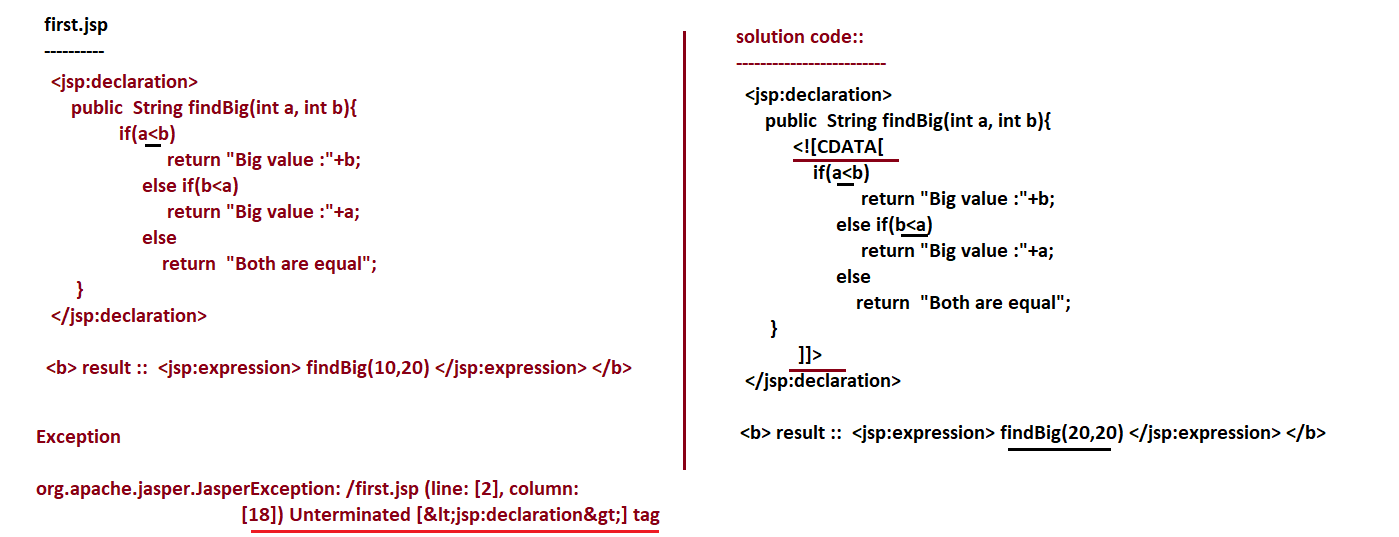
1. **Method Definitations** :- we can place method definitations in declaration tag and those methods can be called from scriplet or expression tag.



**2.Class Definition/Interface Definition:-**  we can place the class definition or interface definition in declaration tag.



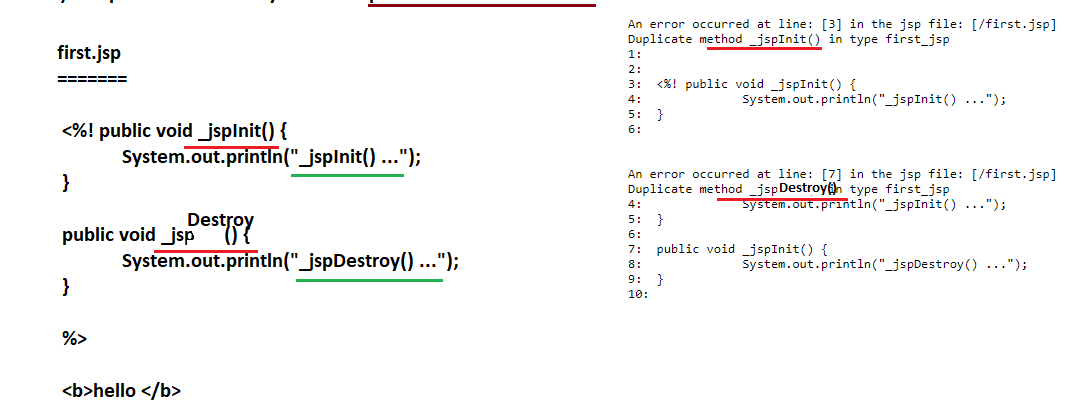
**3.Problem in xml syntax:** In the xml syntax of declaration we have problem with “<” symbol and it can be rectified using <!CDATA[..]]> concept.



4. Can we place \_jspInit(),\_jspDestroy(),\_jspService(-,-) method definitions in declaration tag?

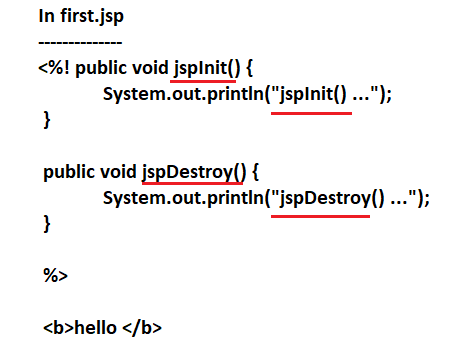
No. not possible because it becomes duplicate methods in JES class. Java supports overloaded methods.. but it does not support duplicate methods.





5. can we place the jspDestroy() , jspInit() method definitions in declaration tag?

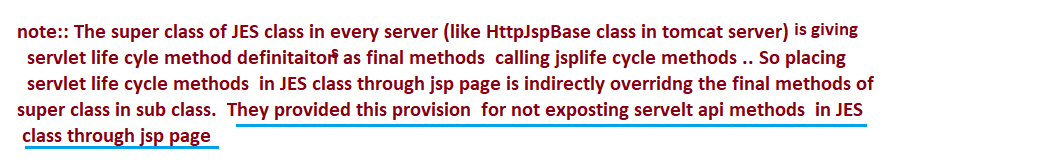
Yes , we can place them because they are not coming as default method definitations in JES class.



**6.** can we place servlet life cycle methods definitations in declaration tags of jsp page?

Not possible because these servlet life cycle methods placed in declaration tags of jsp page goes to JES class and indirectly becomes overriding methods for final servlet life cycle methods available in **HttpJSPBase** Class(the super class of JES class) since we can not override super class final methods in sub classes the error will be generated.





**7.**Important Points:-

=> The variables placed in scriptlet becomes local variables of \_jspService(-,-) method.

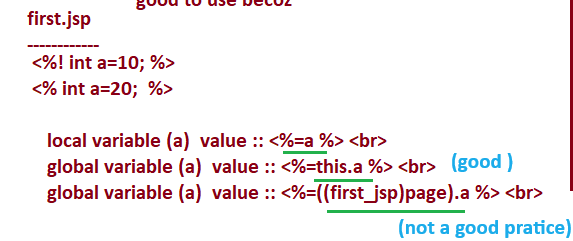
=> The variables placed in declaration tag becomes instance variables of JES class.

=> The classes placed in declaration tag becomes inner classes of JES class.

=> the interfaces placed in declaration tag becomes inner interfaces of JES class.

**8.** if declaration tag variable and scriplet variable both are havng same name then how can we differentiate them in expression tag?

A. using “this” operator or using “page” implicit object.



Note: -

In JES class.

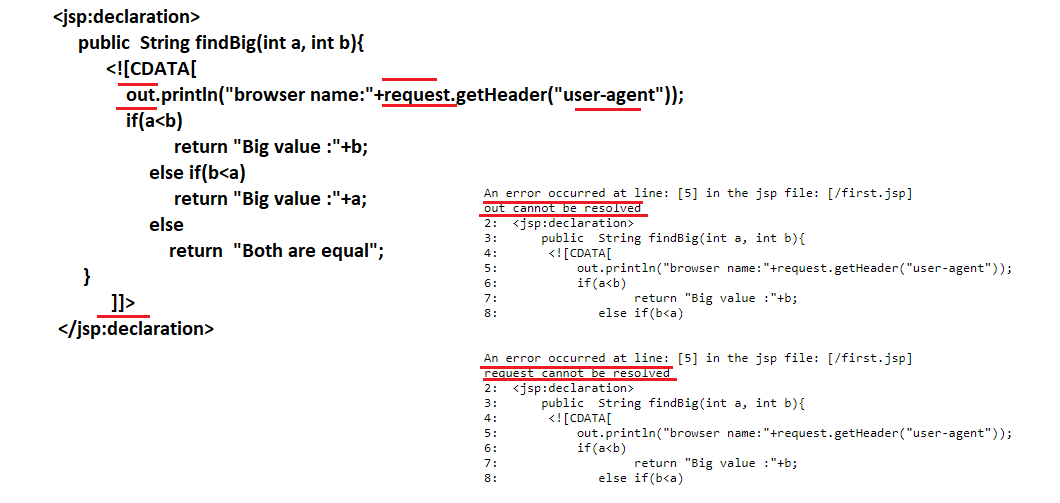
Final java.lang.Object page=this;

Since the implicit object “page” is holding “this” we can use page as alternate “this” but we need to go for some type casting with JES class which changes server to server . Therefore using “this” is good.

9. Implicit Objects with in Declaration Tag:

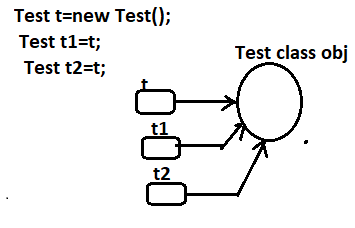
=============================

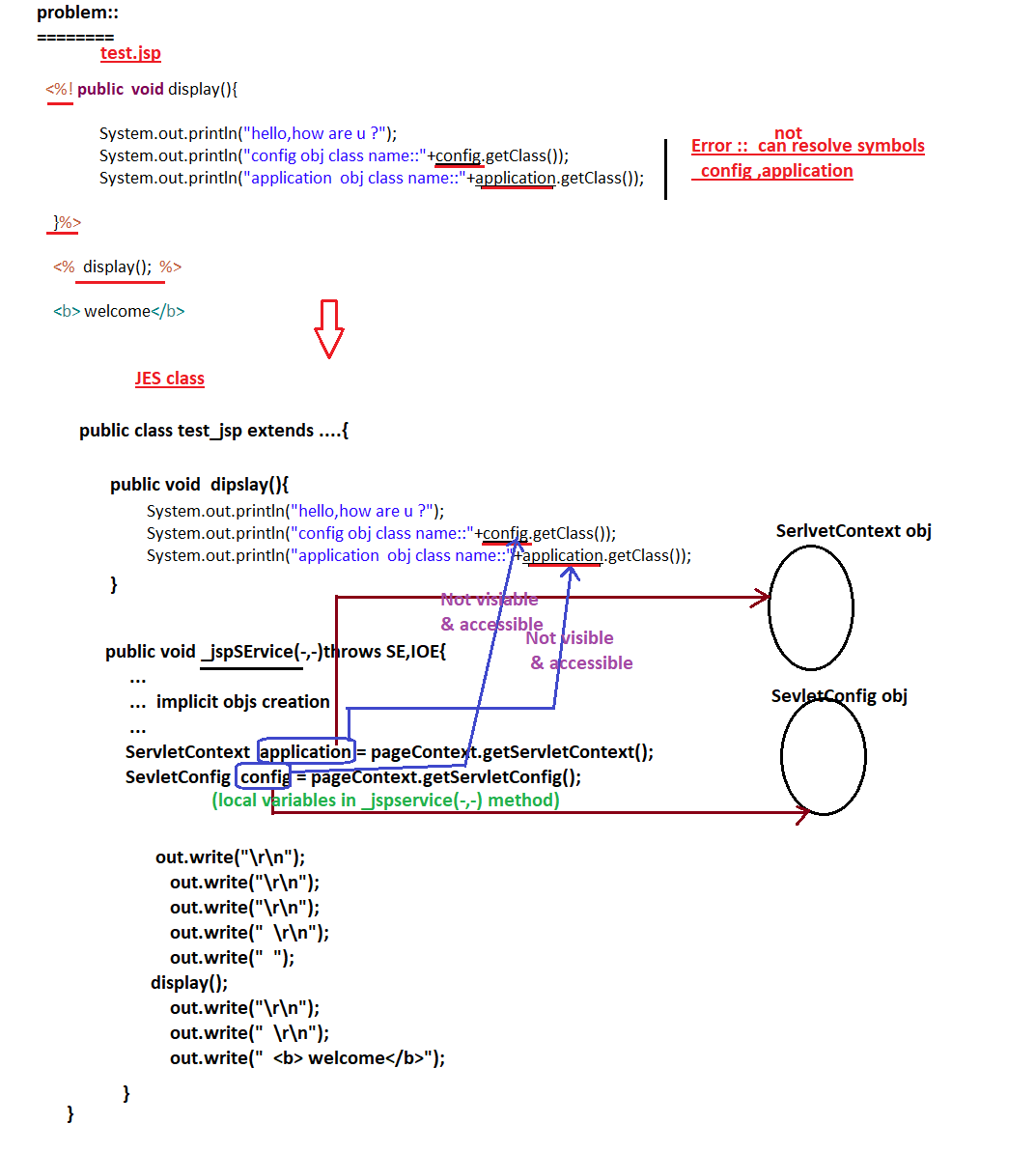
All 9 implicit objs are local variables of \_jspService(-,-) method and code of declaration tag does goes to outside of \_jspService(-,-) method.. so the code of declaration tag can not use implicit objs of jsp page.



Technically speaking the implicit objs are not objects. They are ready made reference variables pointing different objects.

* Application is implicit ref variable referring to ServletContext object.
* Config is implicit ref variable referring to ServletConfig object.
* Out is implicit ref variable referring to JspWriter object and etc..
* We can not use implicit objs in the declaration tags of jsp pages because implicit objs are local variable in \_jspService(-,-) method and declaratinon tag code goes out side of \_jspService(-,-) method in JES class. But we can create additional ref variables for certain objects refered by implicit obj in declaration tags code if needed.

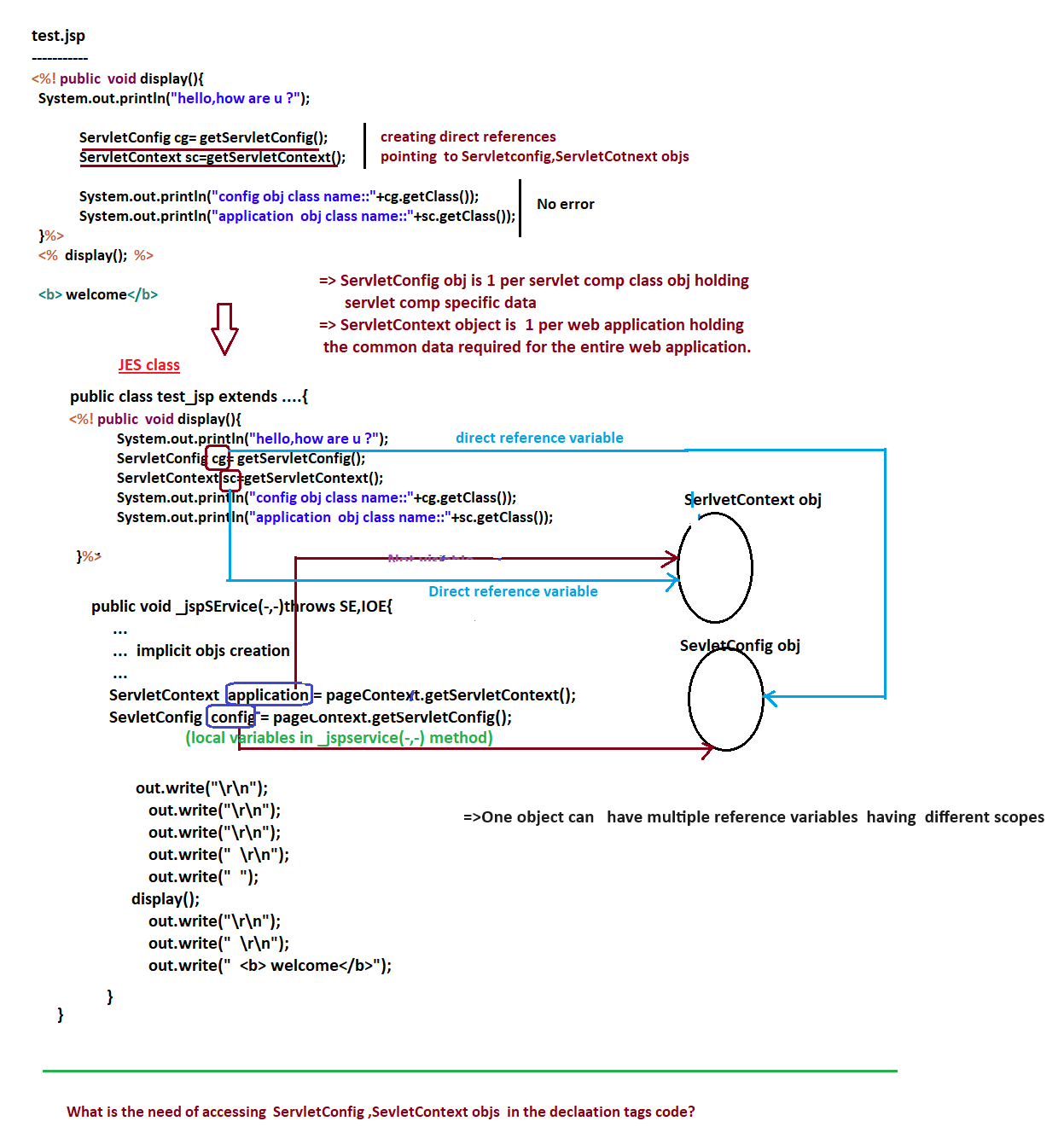




Solution:

=======

Do not use config,application implicit ref variables in declaration tag because they are not visible . So create new set of direct reference variables to ServletConfig , ServletContext objs using servlet api code as show below.



Q) what is need of accessing ServletConfig,ServletContext objs in the declaration tag code?

We can use Servletconfig obj to get init param values and we can use servletContext obj to get Context param values.

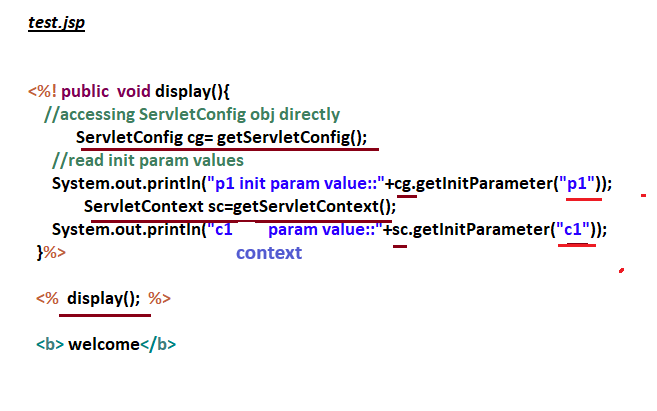
* Init param values are specific to each servlet comp because they allocate memory in servlet Config object.
* Context param values are common for all web comps of web application because they allocate memory in ServletContext object.

Example Code:

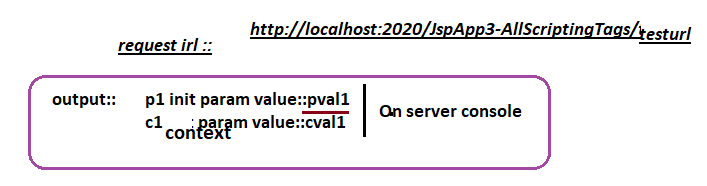
1. Web.xml

=======





Note:- The cfgs done on jsp page in web.xml file will be applied on jsp page only when that JSP page is requested using the url pattern.



**Wrong Thinking:-** in JSP pages we need not to handle exceptions because the JES class will take care of that process.

**Correct Thinking:-** only for the code that goes to \_jspService(-,-) of JES class, we need not to handle the exceptions because the \_jspService(-,-) method of JES class will take care of that, for the remaining code we need to catch and handle the exceptions(for the code placed in the declation tags).

Examples:

