Enumeration is an interface:

An object that implements the  enumeration interface generates a series of elements, one at a time. Successive calls to the  nextelement method return successive elements of the series.

**package** concepts;

**import** java.util.Enumeration;

**import** java.util.Vector;

**public** **class** EnumerationTest {

**public** **static** **void** main(String args[])

{

Enumeration directions;

Vector vobj=**new** Vector();

vobj.add("east");

vobj.add("west");

vobj.add("north");

vobj.add("south");

System.***out***.println(vobj);

directions=vobj.elements();

**while**(directions.hasMoreElements())

{

System.***out***.println(directions.nextElement());

}

}

}

ENUM is a datatype:

**package** concepts;

**import** java.util.Vector;

**public** **class** EnumerationTest {

**public** **enum** day

{

***sunday***,***monday***,***tuesay***,***wednesday***,***friday***,***saturday***

}

**public** **static** **void** main(String args[])

{

System.***out***.println(day.***friday***);

}

}

<html>

<head><title>selector</title>

<style>

.styles

{

background-color:lightblue;

}

#user

{

background-color:yellow;

}

</style>

</head>

<body>

<h1>welcome</h1>

<div class="styles">

entername:<input type="text" name="uname" id="user"><br/>

enterpassword:<input type="password" name="upwd">

</div>

</body>

</html>

--------------------------------------here input type=”text” defines a one input text field

Descendent selector:

The descendant selector matches all elements that are descendants of a specified element.

Child selector: The child selector selects all elements that are the children of a specified element.

Example:

<html>

<head><title>selector</title>

<style>

.styles

{

background-color:lightgreen;

}

#user

{

background-color:yellow;

}

div p

{

background-color:pink;

}

div>h3

{

background-color:violet;

}

</style>

</head>

<body>

<h1>welcome</h1>

<div class="styles">

entername:<input type="text" name="uname" id="user"><br/>

enterpassword:<input type="password" name="upwd"><br/>

<p>paragragh1</p>

<p>paragraph2</p>

<p>paragraph3</p>

<span><p>paragraph4</p></span>

<h3>heading1</h3>

<h3>heading2</h3>

<h3>heading3</h3>

<span><h3>heading4</h3></span>

</div>

</body>

</html>

Class.forName(“oracle.jdbc.driver.OracleDriver”) takes the class name as a string argument and loads it into memory. Second way of loading a class is to simply creating an instance of it using new. Disadvantage of second method is the newly created instance will be of no use if you don't need it.