**9.11 Properties class :-**

**Properties** is a subclass of **Hashtable**. It is used to maintain lists of values in which the key is a **String** and the value is also a **String**. The **Properties** class is used by many other Java classes. For example, it is the type of object returned by **System.getProperties()** when obtaining environmental values.

**Properties** defines the following instance variable:

Properties defaults;

This variable holds a default property list associated with a **Properties** object. **Properties** defines these constructors:

Properties( )   
Properties(Properties *propDefault*)

The first version creates a **Properties** object that has no default values. The second creates an object that uses *propDefault* for its default values. In both cases, the property list is empty.

**Properties** also contains one deprecated method: **save()**. This was replaced by **store()** because **save()** did not handle errors correctly.   
 One useful capability of the **Properties** class is that you can specify a default property that will be returned if no value is associated with a certain key. For example, a default value can be specified along with the key in the **getProperty()** method-such as **getProperty(**"**name**"**,** "**default value**"**)**. If the "name" value is not found, then "default value" is returned. When you construct a **Properties** object, you can pass another instance of **Properties** to be used as the default properties for the new instance. In this case, if you call **getProperty(**"**foo**"**)** on a given **Properties** object, and "foo" does not exist, Java looks for "foo" in the default **Properties** object. This allows for arbitrary nesting of levels of default properties.

The following example demonstrates **Properties**. It creates a property list in which the keys are the names of states and the values are the names of their capitals. Notice that the attempt to find the capital for Florida includes a default value.

**// Demonstrate a Property list.   
import java.util.\*;   
class PropDemo {   
public static void main(String args[]) {   
Properties capitals = new Properties();   
Set states;   
String str;   
capitals.put("Illinois", "Springfield");   
capitals.put("Missouri", "Jefferson City");   
capitals.put("Washington", "Olympia");   
capitals.put("California", "Sacramento");   
capitals.put("Indiana", "Indianapolis");   
// Show all states and capitals in hashtable.   
states = capitals.keySet(); // get set-view of keys   
Iterator itr = states.iterator();   
while(itr.hasNext()) {   
str = (String) itr.next();   
System.out.println("The capital of " +   
str + " is " +   
capitals.getProperty(str)   
+ ".");   
}   
System.out.println();   
// look for state not in list — specify default   
str = capitals.getProperty("Florida", "Not Found");   
System.out.println("The capital of Florida is "   
+ str + ".");   
}   
}**

The output from this program is shown here:

The capital of California is Sacramento.   
The capital of Washington is Olympia.   
The capital of Missouri is Jefferson City.   
The capital of Indiana is Indianapolis.   
The capital of Illinois is Springfield.   
The capital of Florida is Not Found.   
Since Florida is not in the list, the default value is used.

Although it is perfectly valid to use a default value when you call **getProperty( )**, as the preceding example shows, there is a better way of handling default values for most applications of property lists. For greater flexibility, specify a default property list when constructing a **Properties** object. The default list will be searched if the desired key is not found in the main list. For example, the following is a slightly reworked version of the preceding program, with a default list of states specified. Now, when Florida is sought, it will be found in the default list:

**// Use a default property list.   
import java.util.\*;   
class PropDemoDef {   
public static void main(String args[]) {   
Properties defList = new Properties();   
defList.put("Florida", "Tallahassee");   
defList.put("Wisconsin", "Madison");   
Properties capitals = new Properties(defList);   
Set states;   
String str;   
capitals.put("Illinois", "Springfield");   
capitals.put("Missouri", "Jefferson City");   
capitals.put("Washington", "Olympia");   
capitals.put("California", "Sacramento");   
capitals.put("Indiana", "Indianapolis");   
// Show all states and capitals in hashtable.   
states = capitals.keySet(); // get set-view of keys   
Iterator itr = states.iterator();   
while(itr.hasNext()) {   
str = (String) itr.next();   
System.out.println("The capital of " +   
str + " is " +   
capitals.getProperty(str)   
+ ".");   
}   
System.out.println();   
// Florida will now be found in the default list.   
str = capitals.getProperty("Florida");   
System.out.println("The capital of Florida is "   
+ str + ".");   
}   
}**