JAVA.IO.OBJECTINPUTSTREAM.RESOLVEOBJECT METHOD

Description

The **java.io.ObjectInputStream.resolveObject*Objectobj*** method will allow trusted subclasses of ObjectInputStream to substitute one object for another during deserialization. Replacing objects is disabled until enableResolveObject is called. The enableResolveObject method checks that the stream requesting to resolve object can be trusted. Every reference to serializable objects is passed to resolveObject. To insure that the private state of objects is not unintentionally exposed only trusted streams may use resolveObject.

This method is called after an object has been read but before it is returned from readObject. The default resolveObject method just returns the same object.

When a subclass is replacing objects it must insure that the substituted object is compatible with every field where the reference will be stored. Objects whose type is not a subclass of the type of the field or array element abort the serialization by raising an exception and the object is not be stored.

Declaration

Following is the declaration for **java.io.ObjectInputStream.resolveObject** method

```
protected Object resolveObject(Object obj)
```

Parameters

- **obj** -- object to be substituted

Return Value

This method returns the substituted object.

Exception

- **IOException** -- any of the usual Input/Output exceptions.

Example

The following example shows the usage of **java.io.ObjectInputStream.resolveObject** method.

```
package com.tutorialspoint;

import java.io.*;

public class ObjectInputStreamDemo extends ObjectInputStream {

   public ObjectInputStreamDemo(InputStream in) throws IOException {
      super(in);
   }

   public static void main(String[] args) {

      String s = "Hello World";
      try {

         // create a new file with an ObjectOutputStream
         FileOutputStream out = new FileOutputStream("test.txt");
         ObjectOutputStream oout = new ObjectOutputStream(out);

         // write something in the file
         oout.writeUTF(s);
```

```
         oout.flush();

         // create an ObjectInputStream for the file we created before
         ObjectInputStreamDemo ois =
                  new ObjectInputStreamDemo(new FileInputStream("test.txt"));

         // enable object resolving
         ois.enableResolveObject(true);

         // get the class for string and print the name
         System.out.println("" + ois.resolveObject(ois.readUTF()));


      } catch (Exception ex) {
         ex.printStackTrace();
      }

   }
}
```

Let us compile and run the above program, this will produce the following result:

```
Hello World
```