In my previous tutorials we have seen [basics](http://learnfromexamples.com/java-serialization-tutorial-example/) of Java Serialization and [how serialization will behave in terms of inheritance](http://learnfromexamples.com/java-serialization-with-inheritance/), in this tutorial we are going to discuss about difference between Transient variable and Static Variable. Before going to discuss about Transient Variable vs Static Variable, we will discuss about the role of each keyword in Serialization.

[](http://learnfromexamples.com/transient-variable-vs-static-variable-in-java/transient-variable-vs-static-variable/)

Transient Variable vs Static Variable

Transient Variable

Using Transient variable we can restrict the instance variable from Serializing. So transient variable will be ignored during serialization process.

Static Keyword

Serialization will happen only for variables specific to particular instance, but static variables is not specific to any instance. So static variables won’t participate in Serialization process.

Rules for Transient Variable vs Static Variable

1. Transient variable will be ignored during serialization.
2. Static variable won’t even participate in Serialization.
3. Static variable will be serialized if the value is initialized during declaration itself.
4. If a variable contains both transient and static keyword, also if the value is initialized during declaration, then it will be serialized. Because here transient modifier will be ignored and static modifier will take over the actions.
5. Final variable will be serialized.
6. If a variable contains both final and transient keyword, then it will be serialized. Because here transient modifier will be ignored and final modifier will take over the actions.

We will see Transient Variable vs Static Variable along with an example

**TransientStaticVariable .java**

|  |
| --- |
| package com.learnfromexamples.transientvsstatic;    import java.io.Serializable;    public class TransientStaticVariable implements Serializable{          public TransientStaticVariable() {            System.out.println("Constructorr Called...");        }        public transient String variableOne;        public transient String variableTwo = "V2";        public static String variableThree;        public static String variableFour = "V4";        public transient static String variableFive;        public transient static String variableSix = "V6";        final String variableSeven = "V7";        transient final String variableEight = "V8";  } |

**Serialization.java**

|  |
| --- |
| package com.learnfromexamples.transientvsstatic;    import java.io.FileNotFoundException;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.ObjectOutputStream;    public class Serialization {     public static void main(String[] args) throws IOException {   TransientStaticVariable staticVariable = new TransientStaticVariable();   staticVariable.variableOne = "V11";   staticVariable.variableTwo = "V22";   staticVariable.variableThree = "V33";   staticVariable.variableFour = "V44";   staticVariable.variableFive = "V55";   staticVariable.variableSix = "V66";   // since variableSeven and variableEight are final we can't reassign the values   //staticVariable.variableSeven = "V77";   //staticVariable.variableEight = "V88";     FileOutputStream fileOutputStream = new FileOutputStream("transientstatic.ser");   ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);   objectOutputStream.writeObject(staticVariable);   }    } |

**Output**

|  |
| --- |
| Constructorr Called... |

**Deserialization.java**

|  |
| --- |
| package com.learnfromexamples.transientvsstatic;    import java.io.FileInputStream;  import java.io.FileNotFoundException;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.ObjectInputStream;  import java.io.ObjectOutputStream;    public class DeSerialization {     public static void main(String[] args) throws IOException, ClassNotFoundException {     FileInputStream fileInputStream = new FileInputStream("transientstatic.ser");   ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);   TransientStaticVariable staticVariable = (TransientStaticVariable)objectInputStream.readObject();   System.out.println("After Deserialization...");   System.out.println(staticVariable.variableOne);   System.out.println(staticVariable.variableTwo);   System.out.println(staticVariable.variableThree);   System.out.println(staticVariable.variableFour);   System.out.println(staticVariable.variableFive);   System.out.println(staticVariable.variableSix);   System.out.println(staticVariable.variableSeven);   System.out.println(staticVariable.variableEight);   }    } |

**Output**

|  |
| --- |
| After Deserialization...  null  null  null  V4  null  V6  V7  V8 |

**Key Points from the above example.**

* variableOne and VariableTwo were null because as per Rule #1 transient variable will not be serialized.
* variableThree was null because as per Rule #2 static variable won’t participate in Serialization.
* variableFour was “V4” because as per Rule #3 static variable will be serialized only if the value is initialized during declaration.
* variableFive was null because it is both static and transient also its value is not initialized during declaration.
* variableSix was “V6” because as per Rule #4 if variable contains both transient and static keyword, also if that value is initialized during declaration, then it will be serialized.
* variableSeven was “V7” because as per Rule #5 final variable will be serialized.
* variableEight was “V8” because as per Rule #6 if variable contains both final and transient keyword, then it will be serialized.

If you have any questions on this Transient Variable vs Static Variable concept, please post your queries in the comment box. If you like this post, please share it with your friends and also provide your feedback in the in the comment box.