

The serialization runtime associates with each serializable class a version number, called a **serialVersionUID**, which is used during deserialization to verify that the sender and receiver of a serialized object have loaded classes for that object that are compatible with respect to serialization. If the receiver has loaded a class for the object that has a different serialVersionUID than that of the corresponding sender's class, then deserialization will result in an **InvalidClassException**. A serializable class can declare its own **serialVersionUID** explicitly by declaring a field named **serialVersionUID** that must be *static*, *final*, and of type *long*:

```
ANY-ACCESS-MODIFIER static final long serialVersionUID = 1L;
```

- If a serializable class does not explicitly declare a **serialVersionUID**, then the serialization runtime will calculate a default **serialVersionUID** value for that class based on various aspects of the class, as described in the Java(TM) Object Serialization Specification.
- However, it is *strongly recommended* that all serializable classes explicitly declare **serialVersionUID** values, since the default **serialVersionUID** computation is highly sensitive to class details that may vary depending on compiler implementations, and can thus result in unexpected **InvalidClassExceptions** during deserialization.
- Therefore, to guarantee a consistent **serialVersionUID** value across different java compiler implementations, a serializable class must declare an explicit **serialVersionUID** value.
- It is also strongly advised that explicit **serialVersionUID** declarations use the private modifier where possible, since such declarations apply only to the immediately declaring class **serialVersionUID** fields are not useful as inherited members.