BSc (Hons) in Computer Science via GDSE

Advanced API Development

Module Code ITS1114

SERIALIZATION AND DESERIALIZATION

U.W.BAWANTHA BANDARA

2301671056

GDSE67

SUBMISSION DATE(19/7/2024)

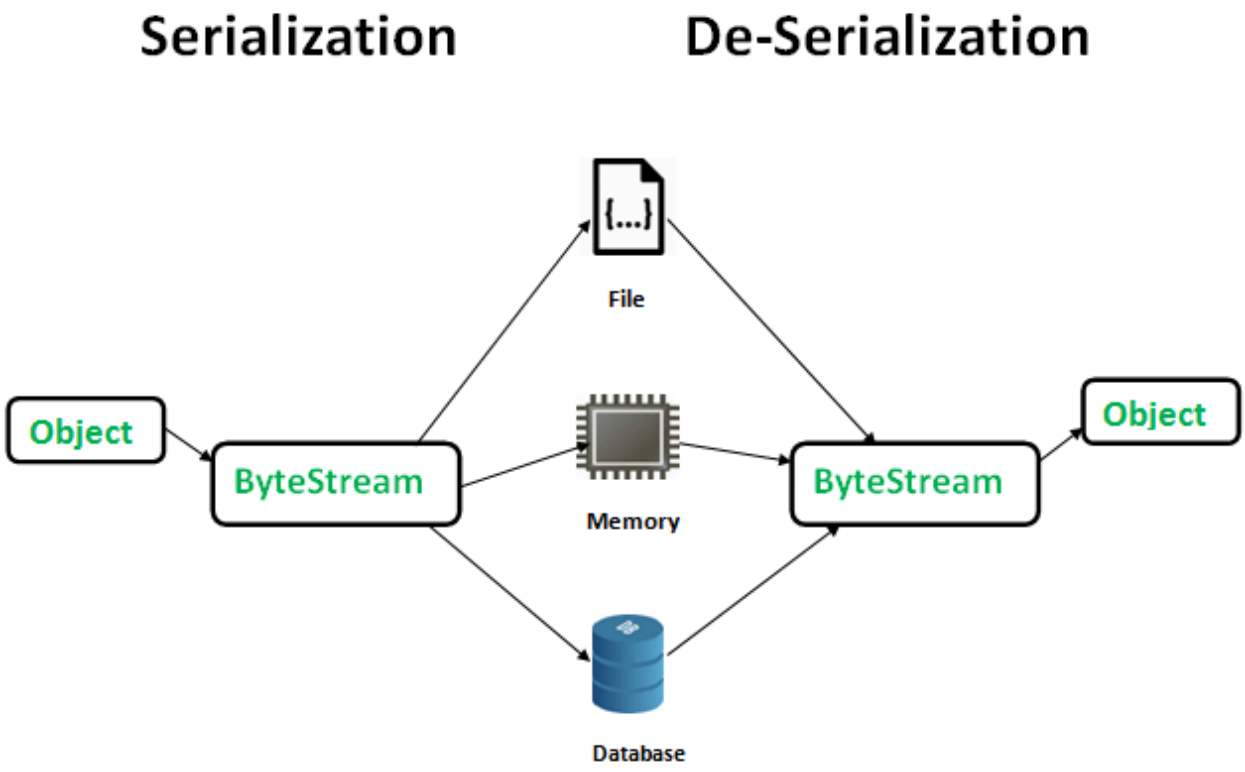
Table of Contents

SERIALIZATION

Serialization is the process of converting objects to ByteSteam.

DESERIALIZATION

Deserialization is the process of converting a Byte stream to an object.it is the reverse process of serialization.



why do we need serialization

- communication
- persistence
- Deep copy

- Caching
- Cross JVM Synchronization

how to serialize an object
the parent class should be implemented by java.io.Serializable interface.
The ObjectOutputStream class contains a writeObject() method for serializing an Object.

```
public final void writeObject(Object obj)
    throws IOException{}
```

The ObjectInputStream class contains a readObject() method for deserializing an object.

```
public final Object readObject()
    throws IOException,
    ClassNotFoundException{}
```

Advantages of Serialization

1. To save/persist the state of an object.
2. To travel an object across a network.

- only non-static data members are saved via serialization process
- constructor of object is never called when an object is deserialized
- Associated objects must be implementing a Serializable interface.

SerialVersionUID The Serialization runtime associates a version number with each Serializable class called a SerialVersionUID, which is used during Deserialization to verify that sender and receiver of a serialized object have loaded classes for that object which are compatible with respect to serialization. If the receiver has loaded a class for the object that has a different UID than that of corresponding sender's class, the Deserialization will result in an InvalidClassException.

A Serializable class can declare its own UID explicitly by declaring a field name. It must be static, final and of type long. i.e- ANY-ACCESS-MODIFIER static final long serialVersionUID=42L; If a serializable class doesn't explicitly declare a serialVersionUID, then the serialization runtime will calculate a default one for that class based on various aspects of class, as described in Java Object Serialization Specification. However it is strongly recommended that all serializable classes explicitly declare serialVersionUID value, since its computation is highly sensitive to class details that may vary depending on compiler implementations, any change in class or using different id may affect the serialized data. It is also recommended to use a private modifier for UID since it is not useful as an inherited member. serialver The serialver is a tool that comes with JDK. It is used to get serialVersionUID numbers for Java classes.