Brandon Justice

SP22-CPSC-35000-003

Dr. Sheikh ‘Sam’ Shamsuddin

November 13th, 2022

**Serialization in Java**

Using Serialization in Java, we can convert an Object to a stream that can transfer the object through a network, saved as a file, or stored within a Data Base and later used within our Java programming application (Pankaj, 2022). *Serialization* is a coding process that converts the state of an object into a byte stream (geeksforgeeks.org, 2021). This bite stream stores all the information about an object (Simpleilearn, 2021). Serialization in Java is one of the essential features of Core Java programming and was introduced as a Core Java feature in Java Development Kit 1.1 (JDK 1.1) (Pankaj, 2022).

The most significant advantage of using Serialization in Java is that it is easy to understand, implement, and customize in Java programming (Pankaj, 2022). Using Serialization in Java provides the developer the mechanism to save or persist the state of an object with the Java program (geeksforgeeks.org, 2021). Another advantage is the ability to move an object across a network (geeksforgeeks.org, 2021), also known as marshaling (Simpleilearn, 2021). Another advantage of using Serialization in Java is that it is independent of Java Virtual Machine (JVM) (Simpleilearn, 2021).

One trivial disadvantage of using Serialization in Java programming Is that it can create security and integrity issues (Pankaj, 2022). According to Joshua Bloch's quote within the levelup.gitconnected.com Article and Senior Software developer “Dev INTJ Code,” today, there is no reason to use Java Serialization (Code, 2020). It is easy to make an object serializable within Java programming, but it can create a detrimental environment as the program gets more prominent (Code, 2020). It can decrease the flexibility to change a class once Serialization, causes an increase in memory requirements, and delay processing when it is implemented, which causes headaches for programmers (Code, 2020). According to Dev INTJ Code, a better option is cross-platform structured-data representations like JSON or Protocol Buffers” (Code, 2020). Serialization is excellent for ease of implementation and customization, but it should be used only within small applications and when needed.

References

Code, D. I. (2020, December 21). *Do not blindly implement the Serializable Interface*. Medium. <https://levelup.gitconnected.com/do-not-blindly-implement-the-serializable-interface-12b596d49687>

geeksforgeeks.org. (2021, October 7). *Serialization and Deserialization in Java with Example - GeeksforGeeks*. GeeksforGeeks. <https://www.geeksforgeeks.org/serialization-in-java/>

Pankaj. (2022, August 3). *Serialization in Java - Java Serialization | DigitalOcean*. Www.digitalocean.com. <https://www.digitalocean.com/community/tutorials/serialization-in-java>

Simpleilearn. (2021, March 30). *Serialization in Java [Advantages and Examples Explained]*. Simplilearn.com. <https://www.simplilearn.com/tutorials/java-tutorial/serialization-in-java>