### **What is Serializable in Android?**

### Serializable is a Java interface that enables an object to be serialized, meaning that it can be converted into a byte stream and stored in a file or transmitted over a network. This is a simple method of saving objects to a file or sending them over a network. When an object is serialized, it is converted into a byte stream that is easily readable by the receiving end. The receiving end can then convert the byte stream back into an object. Serialization is a quick and efficient way of transferring data across activities.

### **What is Parcelable in Android?**

### Parcelable is an Android-specific interface that enables an object to be passed as a parameter from one activity to another. This is a more efficient method compared to serialization, as it doesn’t require the object to be converted into a byte stream. When an object is passed using parcelable, it is passed directly from one activity to another. Parcelable also has the advantage of being able to pass a large amount of data in a single call, making it more efficient than serialization.

### Difference Table:

| **Factor** | **Serializable** | **Parcelable** |
| --- | --- | --- |
| **Overview** | Serializable is the standard Java interface for persistence. | Parcelable is the Android-specific interface for persistence. |
| **Serialization** | Objects are serialized using the Java Serialization API. | Objects are serialized using the Android Parcelable API. |
| **Memory Usage** | Serializable objects are stored in memory and can be retrieved quickly. | Parcelable objects are stored in an Android application bundle and require more time to access. |
| **Speed** | Serializable is slower than Parcelable. | Parcelable is faster than Serializable. |
| **Size** | Serializable objects are larger than Parcelable objects. | Parcelable objects are smaller than Serializable objects. |
| **Implementation** | Serializable objects are implemented by implementing the Serializable interface. | Parcelable objects are implemented by extending the Parcelable class. |
| **Hierarchy** | Serializable supports class hierarchy. | Parcelable does not support class hierarchy. |
| **Reflection** | Serializable objects can be accessed using Java’s reflection API. | Parcelable objects cannot be accessed using Java’s reflection API. |
| **Thread Safety** | Serializable objects are not thread-safe | Parcelable objects are thread-safe |