



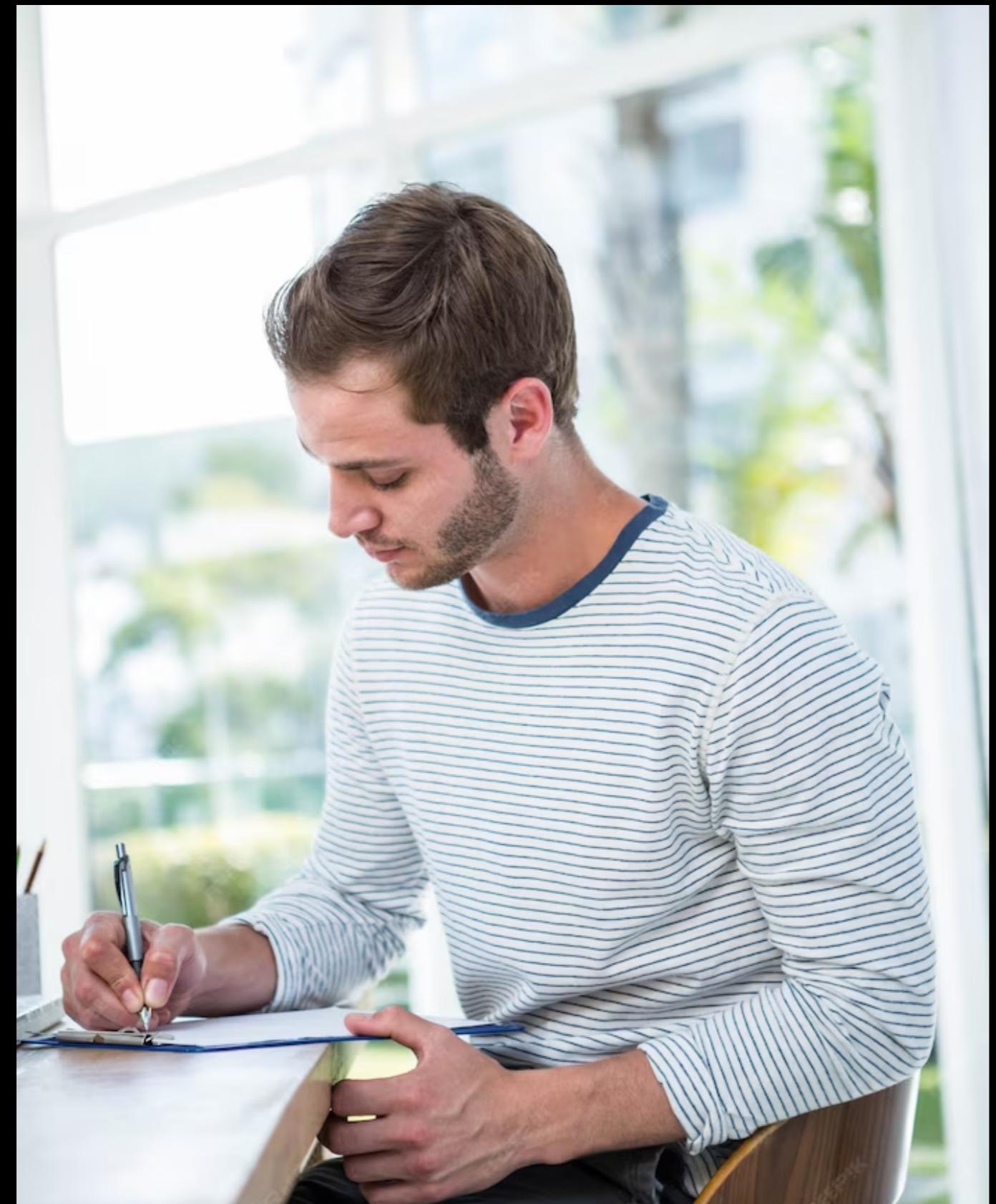
# File Handling in Java: Best Practices and Essential Techniques

By : Janak Raj Joshi



# Mastering File Handling in Java

**File handling** is an essential part of any Java application. It enables you to read, write, and manipulate files and directories. In this presentation, we will cover the **best practices** and **essential techniques** for mastering file handling in Java.



## Understanding File I/O in Java

File I/O in Java involves reading and writing data to and from files. We will cover the **InputStream** and **OutputStream** classes, which are used for reading and writing binary data, as well as the **Reader** and **Writer** classes, which are used for reading and writing character data.



## Working with Files and Directories

Java provides several classes and methods for working with files and directories, including the **File** class, which represents a file or directory path, and the **Path** class, which provides a more flexible way of working with file paths. We will also cover the **java.nio.file** package, which provides additional file system operations.

# Handling Exceptions in File I/O

File I/O operations can throw exceptions, such as **FileNotFoundException** and **IOException**. We will cover how to handle these exceptions using **try-catch** blocks and the **throws** keyword. We will also discuss the importance of closing streams and resources properly.



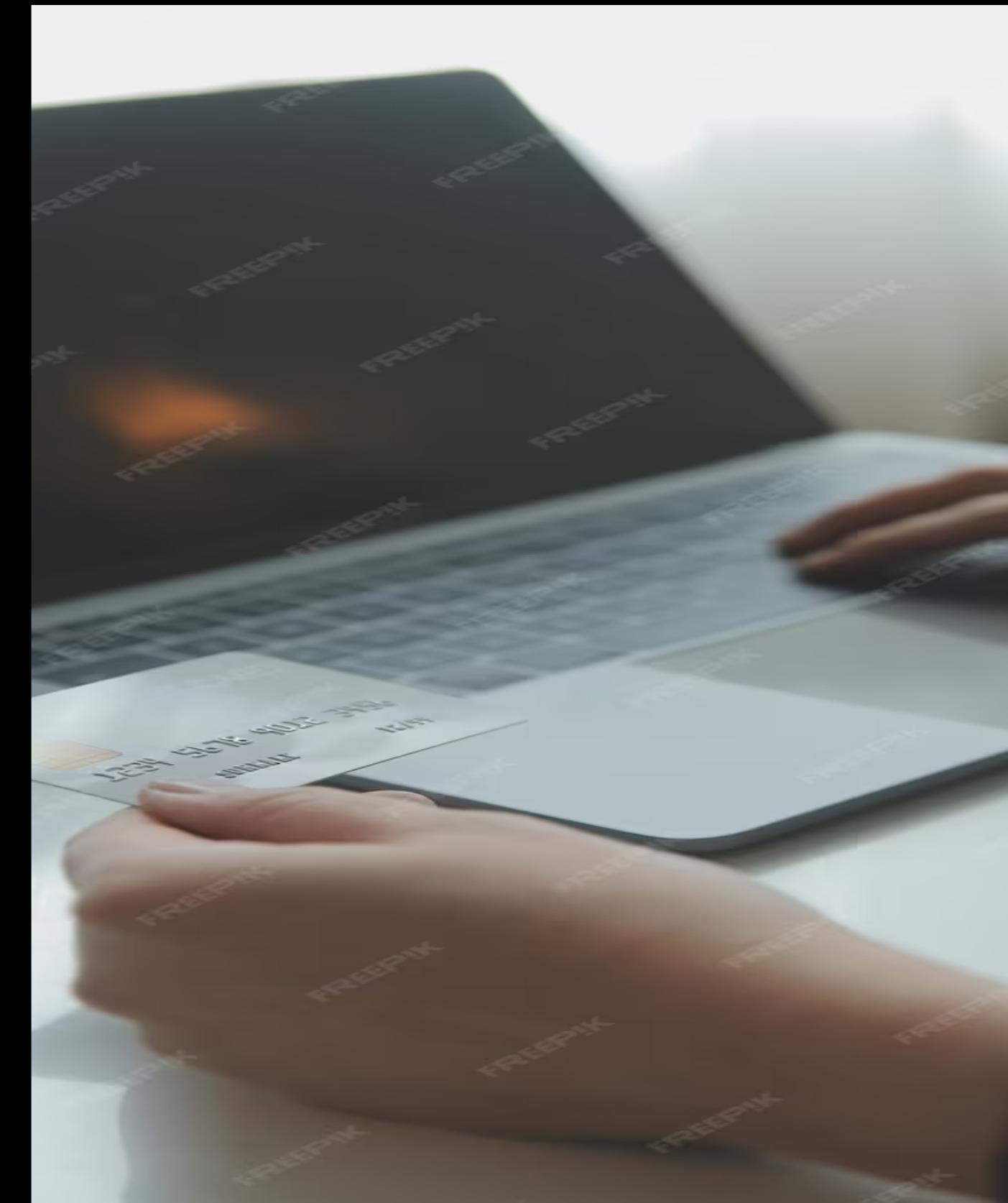
# Reading and Writing Text Files

Text files are a common type of file that can be easily read and written in Java.

We will cover how to use the

**BufferedReader** and **BufferedWriter**

classes for efficient reading and writing of text files. We will also discuss how to handle different character encodings.





# Using Java Serialization

Java Serialization is a mechanism for converting objects into a stream of bytes that can be saved to a file or transmitted over a network. We will cover how to use the **ObjectInputStream** and **ObjectOutputStream** classes for serialization and deserialization of objects.



## Working with Compressed Files

Compressed files, such as **ZIP** and **GZIP** files, are a common way of storing and transmitting large amounts of data. We will cover how to use the **ZipInputStream**, **ZipOutputStream**, **GZIPInputStream**, and **GZIPOutputStream** classes for working with compressed files in Java.

# Conclusion

File handling is a fundamental part of Java programming. By mastering the best practices and essential techniques covered in this presentation, you will be able to work with files and directories in a professional and efficient manner. Thank you for your attention!

Thanks!