try-with-resources

In Java, **try-with-resources** is a feature that automatically closes resources when they are no longer needed. It was introduced in **Java 7** as part of the **AutoCloseable** interface

```
try (ResourceType resource = new ResourceType()) {
        // Use the resource
    } catch (Exception e) {
        e.printStackTrace();
    }
    // The resource is automatically closed here
```

Example: Reading a File with BufferedReader

```
package com.example.exception;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class TryWithResourcesExample {
    public static void main(String[] args) {
        try (FileReader fr = new 
FileReader("d://example.txt");
             BufferedReader br = new BufferedReader(fr)) {


            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
           System.out.println("Exception");
        }
    }
}
```

The BufferedReader is **automatically closed** when the try block ends.

Using AutoCloseable for Custom Resources

If you have a custom resource, implement AutoCloseable:

```
package com.example.exception;
class MyResource implements AutoCloseable {
    public void use() {
        System.out.println("Using resource...");
    }

    @Override
    public void close() {
        System.out.println("Resource closed!");
    }
}

public class CustomTryWithResources {
    public static void main(String[] args) {
        try (MyResource resource = new MyResource())
{
            resource.use();
        }
        // Automatically calls resource.close()
    }
}
```