**8.8- Exercise – DivideException**

**Exercise**

Enhance Exception Handling and Add Custom Logic

**Task**

1. Modify the doDivide(int a, int b) method to print a more descriptive error message in the catch block, such as "Error: Division by zero is not allowed."
2. Add a finally block in the doDivide() method that prints "Division attempt completed."
3. Update the program to validate the input before performing division, ensuring denominator is not 0 before calling doDivide(). If denominator is 0, print "Invalid input: Denominator cannot be zero."

**Hints**

* Ensure that the finally block runs after the try-catch block, regardless of whether an exception is thrown.
* Add an if check before calling doDivide() in the main() method to handle the case when denominator is 0 and skip calling the method.

**Explanation**

This exercise focuses on enhancing exception handling by improving the catch message and ensuring code runs regardless of exceptions with a finally block. Additionally, input validation helps prevent errors before they happen. This practice reinforces robust coding practices and teaches you to write more user-friendly and error-tolerant programs.