import java.util.Scanner;

public class ExceptionHandlingExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter two integers for division:");

try {

int numerator = scanner.nextInt();

int denominator = scanner.nextInt();

int result = divide(numerator, denominator);

System.out.println("Result of division: " + result);

} catch (ArithmeticException e) {

System.out.println("Error: " + e.getMessage());

} catch (java.util.InputMismatchException e) {

System.out.println("Error: Invalid input. Please enter integers.");

} finally {

System.out.println("Program execution completed.");

}

scanner.close();

}

public static int divide(int numerator, int denominator) {

if (denominator == 0) {

throw new ArithmeticException("Division by zero is not allowed.");

}

return numerator / denominator;

}

}import java.util.Scanner;

public class ExceptionHandlingExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter two integers for division:");

try {

int numerator = scanner.nextInt();

int denominator = scanner.nextInt();

int result = divide(numerator, denominator);

System.out.println("Result of division: " + result);

} catch (ArithmeticException e) {

System.out.println("Error: " + e.getMessage());

} catch (java.util.InputMismatchException e) {

System.out.println("Error: Invalid input. Please enter integers.");

} finally {

System.out.println("Program execution completed.");

}

scanner.close();

}

public static int divide(int numerator, int denominator) {

if (denominator == 0) {

throw new ArithmeticException("Division by zero is not allowed.");

}

return numerator / denominator;

}

}

In this program:

1. We use a **try** block to enclose the code that may throw exceptions.
2. Inside the **try** block, we read two integers from the user and attempt to perform division using the **divide** method.
3. We catch two types of exceptions:
   * **ArithmeticException** for division by zero.
   * **InputMismatchException** for invalid input (non-integer input).
4. In the **catch** blocks, we handle the exceptions by displaying error messages.
5. The **finally** block is used to ensure that certain code is executed regardless of whether an exception occurred. In this case, it prints a completion message.

This program demonstrates how to handle exceptions using **try** and **catch** blocks, allowing you to gracefully handle errors and continue program execution.