

Exception

```
class MyCustomException extends Exception {  
    public MyCustomException(String message) {  
        super(message);  
    }  
}  
  
public class ExceptionDemo {  
    // Method that throws a custom checked exception  
    public static void riskyOperation(int value) throws MyCustomException {  
        if (value < 0) {  
            throw new MyCustomException("Value must be non-negative");  
        }  
    }  
  
    // Method to simulate an unchecked exception (ArrayIndexOutOfBoundsException)  
    public static void uncheckedOperation() {  
        int[] array = new int[5];  
        // This will cause ArrayIndexOutOfBoundsException  
        try {  
            int element = array[10];  
        } catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Logging unchecked exception: " + e.getMessage());  
            throw e; // Re-throwing the exception  
        }  
    }  
  
    // Method to simulate an ArithmeticException
```

```
public static void arithmeticOperation() {  
    int zero = 0;  
  
    try {  
        int result = 10 / zero; // This will cause ArithmeticException  
    } catch (ArithmaticException e) {  
        System.out.println("Logging arithmetic exception: " + e.getMessage());  
        throw e; // Re-throwing the exception  
    }  
}  
  
public static void main(String[] args) {  
    try {  
        riskyOperation(-1);  
    } catch (MyCustomException e) {  
        System.out.println("Caught MyCustomException: " + e.getMessage());  
    } catch (Exception e) {  
        System.out.println("Caught general exception: " + e.getMessage());  
    } finally {  
        System.out.println("This block is always executed.");  
    }  
  
    try {  
        uncheckedOperation();  
    } catch (ArrayIndexOutOfBoundsException e) {  
        System.out.println("Caught unchecked exception: " + e.getMessage());  
    }  
  
    try {  
        arithmeticOperation();  
    }
```

```
        } catch (ArithmetricException e) {  
            System.out.println("Caught arithmetic exception: " + e.getMessage());  
        }  
  
        System.out.println("Program continues after exception handling.");  
    }  
}
```

Output

Caught MyCustomException: Value must be non-negative

This block is always executed.

Logging unchecked exception: 10

Caught unchecked exception: 10

Logging arithmetic exception: / by zero

Caught arithmetic exception: / by zero

Program continues after exception handling.