1. Throw and Catch Exception with Try-Catch

Write a Java program that throws an exception and catch it using a try-catch block.

Sample Solution:

Java Code:

```
public class Exception_Example {
  public static void main(String[] args) {
    try {
      int result = divideNumbers(5, 0);
      System.out.println("Result: " + result);
    } catch (ArithmeticException e) {
      System.out.println("Error: " + e.getMessage());
    }
  }
  public static int divideNumbers(int dividend, int divisor) {
    if (divisor == 0) {
      throw new ArithmeticException("Cannot divide the given number by zero!");
    }
    return dividend / divisor;
  }
```

}

Sample Output:

```
Error: Cannot divide the given number by zero!
```

Explanation:

In the above exercise, the divideNumbers method takes two integers as input and checks if the divisor is zero. If the divisor is zero, it throws an ArithmeticException with the message "Cannot divide the given number by zero!"

In the main method, we call the divideNumbers method with 10 as the dividend and 0 as the divisor. Since the divisor is zero, it throws an exception. Using a try-catch block, we catch the exception and print the error message "Error: Cannot divide by zero".

