**I created the following modules for a homework assignment:**

## def raiseIndexError()

**This function raises an IndexError.**

## def raiseZeroDivisionErrorWithMessage(message:str)

**This function raises a ZeroDivisionError with the specified message.**

## def raiseThisException(exception)

**This function will receive an exception as an argument. It should raise the exception it receives.**

## def catchAndReturnMessage(message:str, main\_function:callable) -> str:

**This function will receive a string message and a callable as an argument. The callable has a chance of raising an Exception. This function should invoke the callable, if the callable does not raise an exception it should return the message it received as an argument. If the callable raises an exception, it should handle the exception and return the message of the raised exception.**

## def catchCleanupAndThrow(main\_supplier:callable, index\_supplier:callable, zero\_supplier:callable, cleanup:callable) -> str:

**This function will receive four callables as arguments. The function should invoke the main\_supplier, main\_supplier will return a string, if main\_supplier does not raise an exception this function will return the value returned by main supplier. main\_supplier has the possibility of raising an IndexError and a ZeroDivisionError. If main\_supplier raises an IndexError this function will call the index\_supplier function and return the value returned by index\_supplier. If main\_supplier raises a ZeroDivisionError this function will call the zero\_supplier function and return the value returned by zero\_supplier. Before the function returns regardless if an exception is thrown or not this function should invoke the cleanup function.**