Week 6 - Assignment (2 marks)

Note: All programs must use the appropriate C++ features.

Objective: Develop a C++ application that performs vector operations. This program will enable vector multiplication and division using functors and lambda expressions for element-wise calculations. It should also robustly handle invalid operations such as differing vector sizes and division by zero through both custom and standard exceptions.

Requirements:

1. **NewVector** Class
    a. **Data Storage**: Store data using either a vector or array**.**
    b. **Operations Support**: Implement multiplication and division using functors. Create specific classes for these operations to encapsulate the functionality.
    c. **Operators Overloading**: Overload the operators *, /, and << to facilitate vector operations.
    d. **Constructors**:
        i. Default Constructor: NewVector(int num_elem=10). Initialize the NewVector with random values using lambda expressions.
        ii. Parameterized Constructor: Initialize the NewVector with provided input values, e.g, a vector or array.
2. Custom Exception Class
    a. Define a VectorOperationException class to address specific vector operation errors such as mismatched dimensions and zero-division.
3. Exception Handling
    a. Implement try-catch blocks to efficiently manage exceptions during vector operations.

4. Modify the given main() function to achieve the following features:
    a. User Interface: Design an interface that continuously interacts with the user until they opt to quit, gracefully handling incorrect data entries.
    b. Input Handling: Enable users to specify the number of elements for two NewVectors, with an option to initialize using the default constructor or manually input values.
    c. Operation Demonstrations: Show successful vector operations and handle exceptions where operations fail, such as mismatched dimensions or zero division. Therefore, there are at least three test cases: one successful case, one error case due to mismatched dimensions, and one error case due to zero division.

The input and output should be like below:

Input & Output:

Modify the given week6.cpp to achieve the result like below.

You should have one successful case, one error case due to mismatched dimensions, and one error case due to zero division.

```
Demo for vector operations
Enter the number of elements for the vector: 2
Mannually input values of this vector (Y/N): y
Enter a number #1: 1
Enter a number #2: 1
Vector vec1:
1 1
Enter the number of elements for the vector: 2
Mannually input values of this vector (Y/N): y
Enter a number #1: 0
Enter a number #2: 1
Vector vec2:
0 1
Result of vec1 * vec2:
0 1
Vector operation error: Division by zero encountered.
Quit (Y/N): n
Enter the number of elements for the vector: 3
Mannually input values of this vector (Y/N): n
Vector vec1:
6.2 6.4 0.5
Enter the number of elements for the vector: 3
Mannually input values of this vector (Y/N): n
Vector vec2:
4.5 8.1 2.7
Result of vec1 * vec2:
27.9 51.84 1.35
Result of vec1 / vec2:
1.37778 0.790123 0.185185
Quit (Y/N): test
Invalid input. Quit (Y/N): n
Enter the number of elements for the vector: 4
Mannually input values of this vector (Y/N): n
Vector vec1:
9.1 0.4 0.2 5.3
Enter the number of elements for the vector: 3
Mannually input values of this vector (Y/N): n
Vector vec2:
9.2 8.2 2.1
Vector operation error: Vectors must be of the same size.
Quit (Y/N): y
PS H:\Courses\2800ICT\workspace>
```

Submit:

1, **all C++ source code**: *.cpp and *.hpp if your code is organized into separate files.

Organizing the source code into separate files is not mandatory.

You can consolidate all code into a single cpp file.

2, **week6.txt**: a txt file contains all the source code.

3, **output.jpg**, or output.png, or output.bmp: a screenshot of the output by your program

Please refer to the submission page for the Marking Rubric.