Nithin S
221IT085

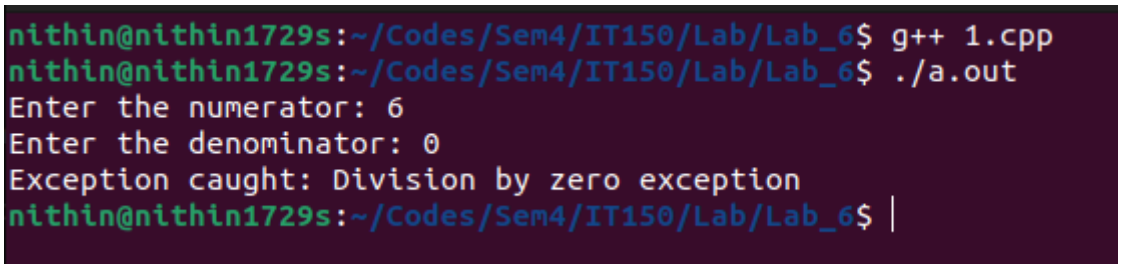
IT150 Lab Assignment 6

1. Write a function which throws a division by zero exception and catch it in catch block. Write a C++ program to demonstrate usage of try, catch and throw to handle exception.

CODE

```cpp
#include <iostream>

using namespace std;


double divide(int numerator, int denominator) {
    if (denominator == 0) {
        throw "Division by zero exception";
    }
    return static_cast<double>(numerator) / denominator;
}

int main() {
    int numerator, denominator;
    cout << "Enter the numerator: ";
    cin >> numerator;
    cout << "Enter the denominator: ";
    cin >> denominator;

    try {
        double result = divide(numerator, denominator);
        cout << "Result of division: " << result << endl;
    } catch (const char* exception) {
        cerr << "Exception caught: " << exception << endl;
    }

    return 0;
}
```

OUTPUT



2. Write a Program to:

(i) Demonstrate the Catching of All Exceptions.

(ii) which handles array of bounds exception

(iii) implement the exception handling with multiple catch statements.

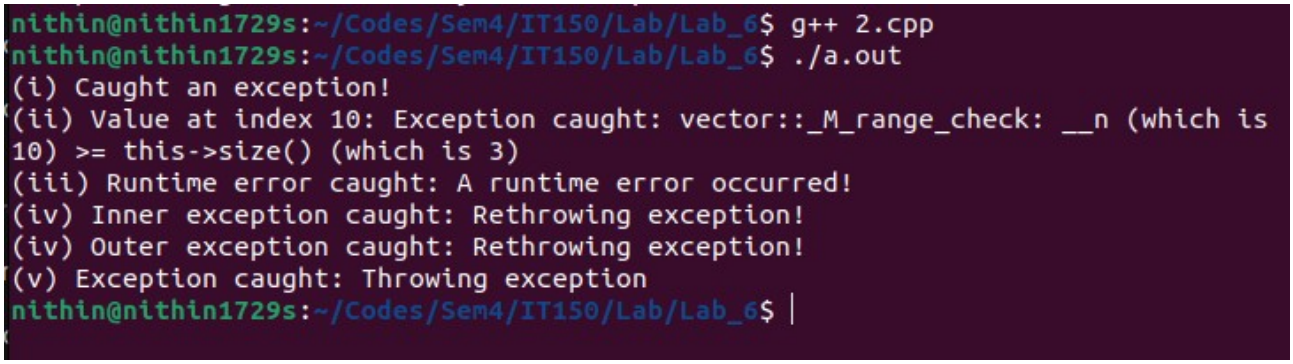
(iv) implement the exception handling with re-throwing in exceptions

(v) implement the exception handling with the functionality of testing the throw restrictions.

CODE

```cpp
#include <iostream>
#include <vector>

using namespace std;

int main() {
    // (i) Catching all exceptions
    try {
        throw "An exception occurred!";
    } catch (...) {
        cout << "(i) Caught an exception!" << endl;
    }

    // (ii) Handling array out of bounds exception
    try {
        vector<int> arr = {1, 2, 3};
        cout << "(ii) Value at index 10: " << arr.at(10) << endl;
    } catch (const out_of_range& e) {
        cerr << "Exception caught: " << e.what() << endl;
    }

    // (iii) Multiple catch statements
    try {
        throw runtime_error("A runtime error occurred!");
    } catch (const runtime_error& e) {
        cerr << "(iii) Runtime error caught: " << e.what() << endl;
    } catch (const exception& e) {
        cerr << "(iii) Exception caught: " << e.what() << endl;
    }

    // (iv) Re-throwing exceptions
    try {
        try {
            throw runtime_error("Rethrowing exception!");
        } catch (const runtime_error& e) {
            cerr << "(iv) Inner exception caught: " << e.what() << endl;
            throw; // Re-throwing the caught exception
        }
    } catch (const runtime_error& e) {
        cerr << "(iv) Outer exception caught: " << e.what() << endl;
    }

    // (v) Testing throw restrictions
    try {
        throw runtime_error("Throwing exception");
    } catch (const runtime_error& e) {
        cerr << "(v) Exception caught: " << e.what() << endl;
    }

    return 0;
}
```

OUTPUT


nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_6$ g++ 2.cpp
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_6$ ./a.out
(i) Caught an exception!
(ii) Value at index 10: Exception caught: vector::_M_range_check: __n (which is
10) >= this->size() (which is 3)
(iii) Runtime error caught: A runtime error occurred!
(iv) Inner exception caught: Rethrowing exception!
(iv) Outer exception caught: Rethrowing exception!
(v) Exception caught: Throwing exception
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_6$