# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 9

1) Exception handling

**C++ CODE:**

#include <iostream>

using namespace std;

int main(){

    int divisor, dividend;

    cout << "Enter dividend: ";

    cin >> dividend;

    cout << "Enter divisor: ";

    cin >> divisor;

    try{

        if (divisor == 0)  {

            throw runtime\_error("Division by zero error");

        }

        cout << "Result: " << dividend / divisor << endl;

    }

    catch (runtime\_error &e){

        cout << "Exception caught: " << e.what() << endl;

    }

    cout << "Press Enter to continue...";

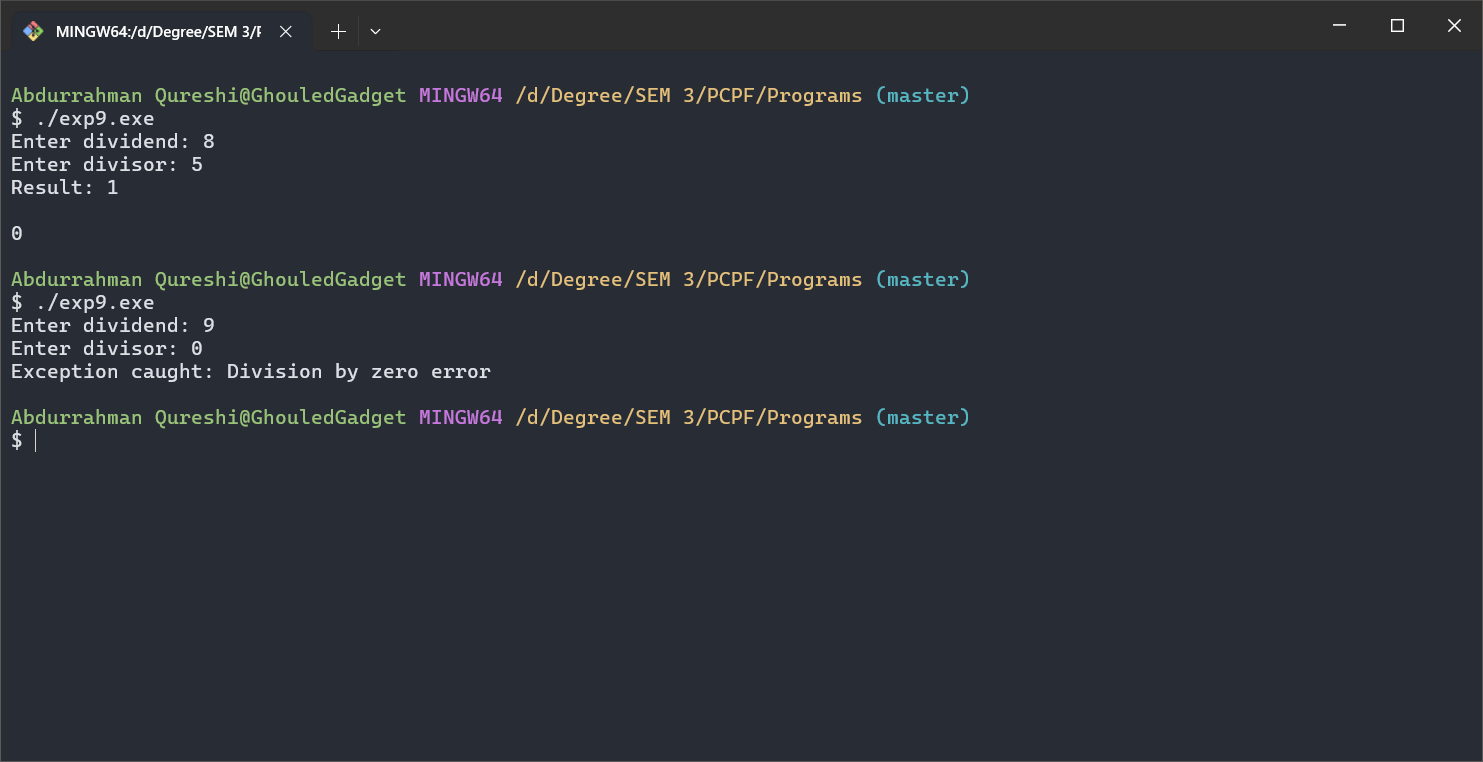
    cin.ignore();

    cin.get();

    return 0;

}

**OUTPUT:**



JAVA CODE:

public class exp9 {

    public static double divide(int numerator, int denominator) throws IllegalArgumentException {

        if (denominator == 0) {

            throw new IllegalArgumentException("Division by zero is not allowed.");

        }

        return (double) numerator / denominator;

    }

    public static void main(String[] args) {

        int num = 10;

        int denom = 0;

        try {

            double result = divide(num, denom);

            System.out.println("Result: " + result);

        } catch (IllegalArgumentException e) {

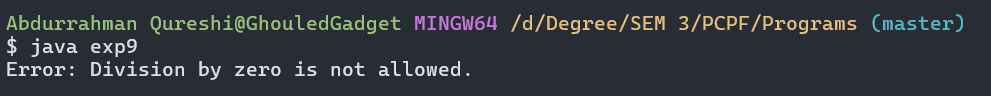
            System.err.println("Error: " + e.getMessage());

        }

    }

}

**OUTPUT:**



2) Garbage Collection.

**C++ CODE:**

#include <iostream>

class MyClass {

public:

    MyClass() { std::cout << "Constructor called!" << std::endl; }

    ~MyClass() { std::cout << "Destructor called!" << std::endl; }

};

int main() {

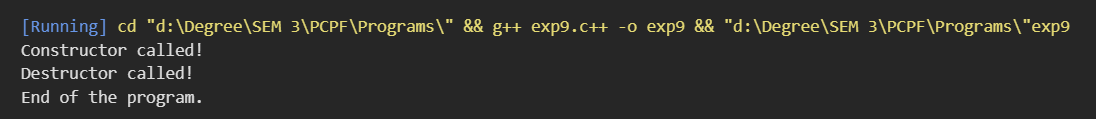
        MyClass obj = MyClass();

    std::cout << "End of the program." << std::endl;

    return 0;

}

**OUTPUT:**



JAVA CODE:

public class exp9 {

    static class MyClass {

        MyClass() {

            System.out.println("Constructor called!");

        }

    }

    public static *void* main(*String*[] *args*) {

*MyClass* obj = new MyClass();

        System.out.println("Auto destructor called.");

    }

}

**OUTPUT:**

