

```
/*This programme is an example of how try-catch-throw are used to handle exceptions*/

#include <iostream>
using namespace std;

void main() {

    int counts[8]{ 5,2,6,0,3,6,1,4 };

    double Inp{ 60 };

    int Out{};
    // The loop divides 60 by every number in the array "counts"
    for (unsigned int i = 0; i < 8; i++)
    {
        try { // This block might have a run-time error
            if (counts[i] == 0) throw "Error division by 0"; // This is exception 1
            (the name of the exception is the message)
            if (counts[i] == 1) throw 20; // This is exception 2 (This naming is
            usefull when having a documnet listing possible errors)
            Out = Inp / counts[i];
            cout << Out << "\n";
        }

        catch (const char aMessage[]) // This block is called when exception 1 occurs
        {
            cerr << "Exception 1 is thrown" << "\n"; // Notice the use of cerr for
            displaying errors
        }

        catch (const int n) // This block is called when the exception 2 occurs
        {
            cerr << "Exception 2 is thrown" << "\n"; // Notice the use of cerr for
            displaying errors
        }

        catch (...) // This is a general catch block called for anything else (not
        recommended)
        {
            cerr << "Another exception is thrown" << "\n"; // Notice the use of cerr
            for displaying errors
        }

    }

    system("pause");
}
```