ExceptionHandlingDemo

1. Open the file ExceptionHandlingDemo - starter and extract the project.
2. Run the project, which should be on Example 1: C++ anything goes exception.
3. Enter age = 22. What happens? Enter age = 14. What happens? Nothing.
4. Let’s add a custom exception. Put a try block around all of the code. Then add an else to the if. Put this code in the else: throw (age);
5. If we have a try, we need a catch. We are throwing an int, so w4e have to catch an int. Add a catch block directly after the end of the try block. It will look like this:

catch (int myNum) {

cout << "Access denied - You must be at least 18 years old.\n";

cout << "Age is: " << myNum<<endl << endl;

}

1. Run the project again. What happens? Information is passed to the user.
2. Comment out the code we just used and un-comment the code for 2.Vector OutOfRange Exception.
3. Now run this part of the program and see what happens. Yes, we crash.
4. Let’s debug and see where we crashed. Put a break point before int length{0}; and start debugging. Enter 5 when we are asked to enter a length.
5. And there we go, at the line vec.at(8)= 7; Why did this crash?
6. Let’s add a try - catch. Put the try before the line vector<int> vec( length);
7. And finish the block after vec.at(8) = 7;
8. Add the catch immediately after.

catch (out\_of\_range exc)

{

cout << "\n Out of range exception caught" <<endl

<< exc.what() << endl;

}

1. Out\_of\_range is a std C++ error that derives from logic\_error and then from exception.
2. Run your project and what happens? You also see that there is an invalid vector subscript.
3. Comment out the code we just used and un-comment the code for 3.Divide By Zero Exception.
4. Run the project.
5. Run the project and enter two integers include one that will cause a divide by 0. What happens? Nothing. You get inf.
6. What is that? he two values of infinity are special values that can be represented by the 32-bit float type or the 64-bit double. Mathematically, infinity is a concept that represents a value greater than any other real number (positive infinity), or smaller than any other real number (negative infinity). You can also generate a positive or negative infinity value as a result of dividing a positive or negative number by zero. Doing these calculations does not result in an exception.

**float** posInfinity = 1.0f / 0;

**float** negInfinity = -1.0f / 0;

Similarly, we have NaN for Not a Number. NaN, acronym for “Not a Number” is an exception which usually occurs in the cases when an expression results in a number that can’t be represented. For example the square root of negative numbers. And the result of dividing zero by zero is [NaN](https://docs.microsoft.com/en-us/dotnet/api/system.double.nan?view=netcore-3.1).

1. DivisionByZero: Historically, dividing a floating-point value by zero doesn't throw an exception; it results in positive infinity, negative infinity, or not a number. It is a mathematical error and we can handle using an exception so that we can gracefully recover from it. If you write a code without using exception handling then the output of division by zero will be shown as infinity which cannot be further processed.
2. Let’s do that by adding a try just before the double result = and have its block end just after the next line, cout << "The quotient is:
3. Put a catch just after the end of the try and put this code inside its block: cout << "\n exception occurred: " << divideByZeroException.what() << endl;
4. The last line of the while block, cout << "\n Enter two numbers, comes right afterward.
5. It looks like we need to write a new exception. So right-click on theHeaderFiles section of the Solution Explorer and add a class.
   1. Don’t include the .cpp part of the class.
   2. Add the include guards:
   3. #ifndef \_DIVIDE\_BY\_ZERO\_H
   4. #define DIVIDE\_BY\_ZERO\_H
   5. #endif
   6. And the class code:

class DivideByZeroException : public runtime\_error

{

public:

DivideByZeroException() : runtime\_error("Attempted to divide by zero") {}

};

1. Run your program. Try to divide by 0. What happens?
2. Now comment out all of what we did and uncomment 4. Catch Any Exception. We are back with being old enough.
3. Run your project, enter 33, then try it again with 15, and what happens? Nothing. Lets try thowing another exception.
4. Put a try block around all of the code.
5. Add an else whose code says throw 505;
6. Add a catch directly after the end of the try block. Make the exception in parentheses look like this: catch (...)
7. The code in the catch block will say:cout << "Access denied - You must be at least 18 years old.\n";
8. Run your project. What happens?