**C# Homework 06**

**Question 1**

What is an exception?

**Answer**

An exception is a problem that arises during the execution of a program. Exception provide a way to transfer control from one part of a program to another. C# exception handling is built upon four keywords: try, catch, finally, and throw.

**Question 2**

What happens in a try block if the program executes without errors?

**Answer**

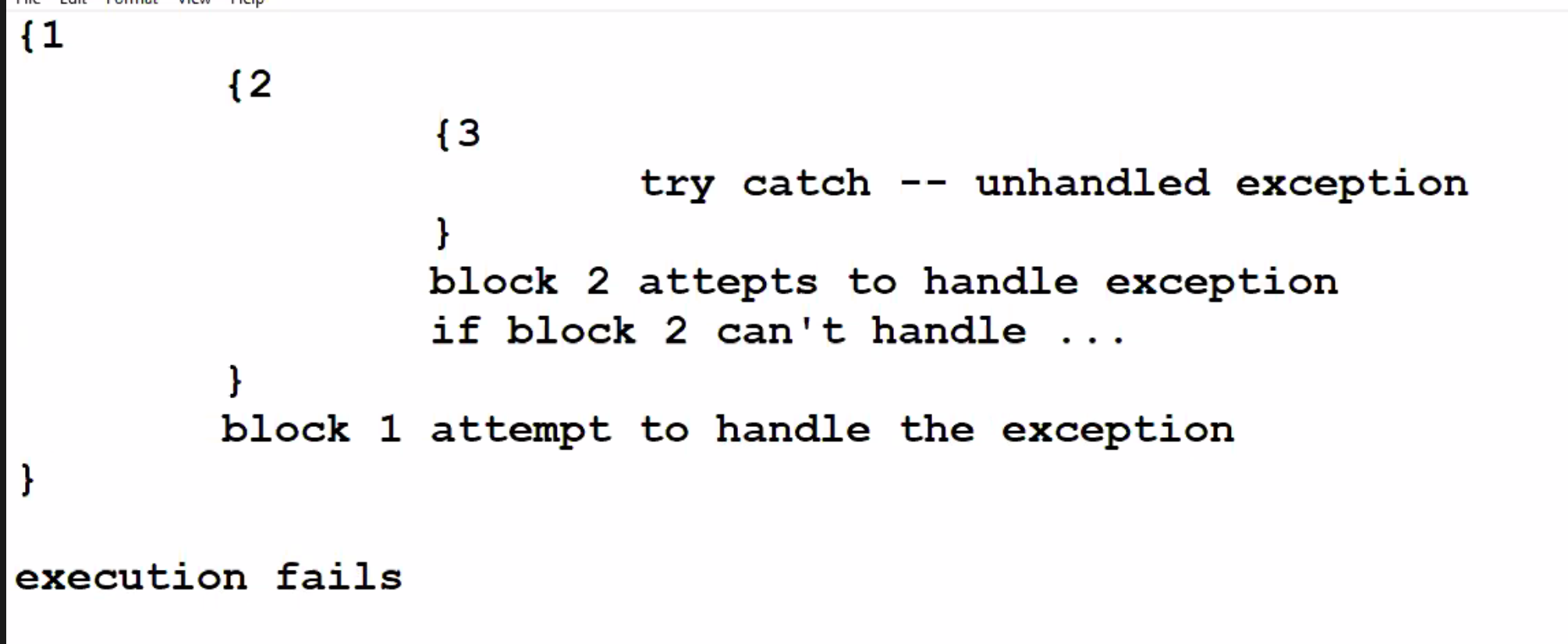
First, the code in try is executed. If there were no errors, then catch(err) is ignored: the execution reaches the end of try and goes on, skipping catch. If an error occurs, then the try execution is stopped, and control flows to the beginning of catch(err).

**Question 3**

How does the catch mechanism work for unhandled exceptions?

**Answer**

An unhandled exception occurs when the application code does not properly handle exceptions. When you try to open a file on a disk, it is common problem for the file to not exist. The .NET Framework will then throw a FileNotFoundException. This is a simple example of a potential known problem that is accounted for within code. If the catch can not handle the exception it throws it up to the next higher up catch statement.



**Question 4**

What happens in a program if an exception block fails to handle a particular error?

**Answer**

You can put a try catch inside the catch block, or you can simply throw the exception again. Its better to have finally block with your try catch so that even if an exception occurs in the catch block, finally block code gets executed. If it can not fix the exception on the last catch the program will terminate.

**Question 5**

What is the parent class for all exceptions? How does this work?

**Answer**

The System. SystemException class is the base class for all predefined system exception.

**Question 6**

How do you determine the type of an error?

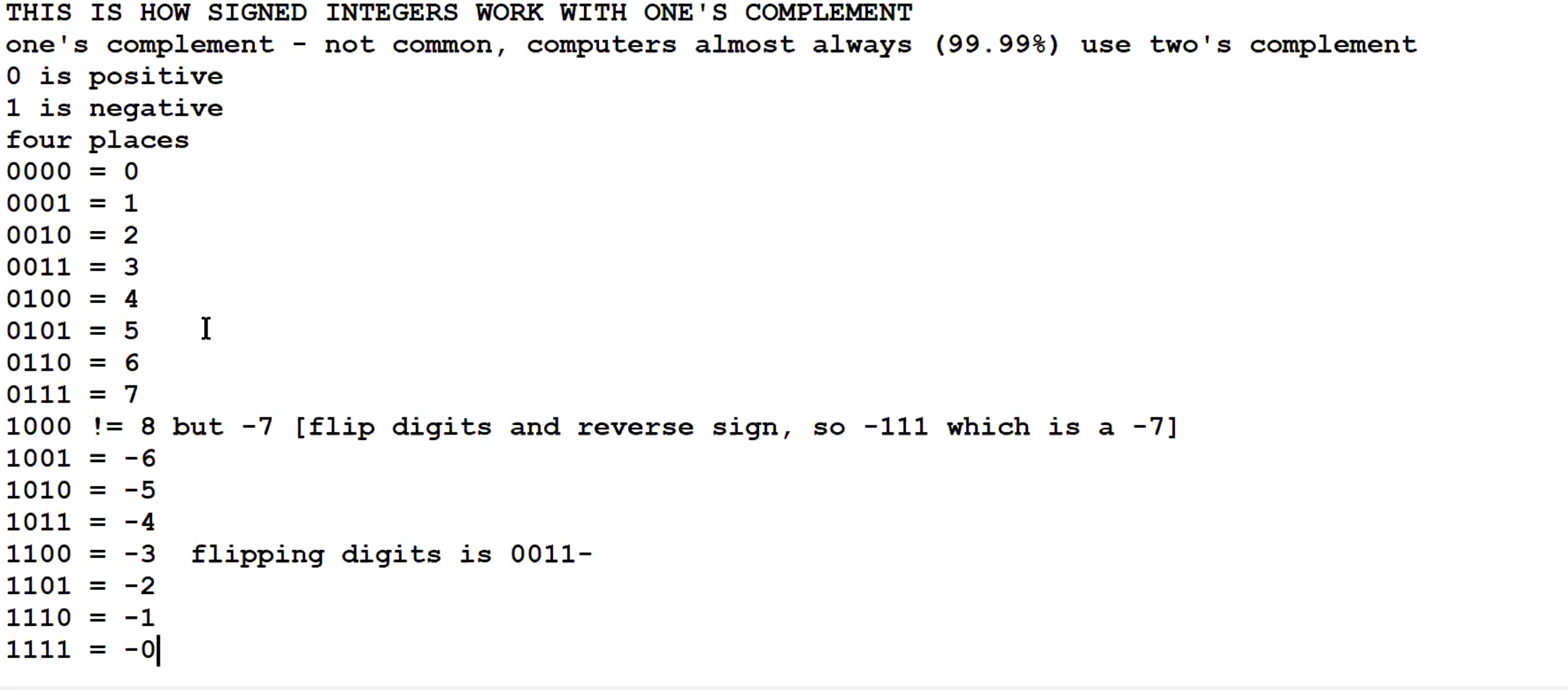
**Answer**

The Exception.Message property gets a message that describes the current exception.

**Question 7**

What is the purpose of integer checking?

**Answer**



**Question 8**

What is the range of values than a signed Int32 type can contain? State the lowest value and the highest value.

**Answer**

Lowest value -2,147,483,648

Highest value 2,147,483,647

**Question 9**

What is the range of values that an unsigned Int32 type can contain? State the lowest value and the highest value. What is the difference between a signed integer and an unsigned integer? Can signed integers and unsigned integers represent the same amount of numbers?

**Answer**

Lowest value 0

Highest value 4,294,967,295

In laymen’s terms an unsigned int is an integer that can not be negative and thus has a higher range of positive values that it can assume. A signed int is an integer that can be negative but has a lower positive range in exchange for more negative values it can assume.

**Question 10**

What does the finally block do?

**Answer**

By using the finally block, you can clean up any resources that are allocated in a try block, and you can run code even if an exception occurs in the try block. Typically, the statements of a finally block run when control leaves a try statement. The finally block always runs.

**Question 11**

When would you not use a finally block in a try/catch construction?

**Answer**