Exception handling is the process of responding to exceptions when a computer program runs. An exception occurs when an unexpected event happens that requires special processing. Examples include a user providing abnormal input, a file system error being encountered when trying to read or write a file, or a program attempting to divide by zero.

Exception handling attempts to gracefully handle these situations so that a program (or worse, an entire system) does not crash. Exception handling can be performed at both the software (as part of the program itself) and hardware levels (using mechanisms built into the design of the CPU).

Example of exception handling in JavaScript

```
try {

  console.log(test);

} catch (err) {

  console.log("Error encountered: " + err);

  console.log("Continuing with the rest of our program…");

}
```

Here, console.log(test) tells the program to print the value of a variable named "test" to the console. However, we have not yet defined this variable, so the console.log method generates an error.

Normally, this would cause the program to crash, but here we have enclosed the error-causing code in a try-catch statement. The program "tries" to run console.log(test), and if it works, the catch block is skipped. But if it causes an error, the error is "caught" — instead of the program crashing, the catch block is executed. Our program produces the output:

```
Error encountered: ReferenceError: test is not defined

Continuing with the rest of our program.
```