

The Exception Class

This lesson discusses how you can use the Exception class for exception handling using various coding scenarios.

We'll cover the following ^

- Components
- Methods

PHP has a class-based exception handling mechanism. The `Exception` class is a built-in class with various methods and properties. In this lesson, we will be just be concerned with learning the basics of exception handling using the PHP `Exception` class.

Components

- `try`: It is the block of code in which exception may arise.
- `catch`: It is the block of code that will be executed when a particular exception is *thrown*.
- `throw`: It is used to *throw* an exception. It is also used to list the exceptions that a function throws, but doesn't handle itself.
- `finally`: This is the block of code that executes at the end once the exception is thrown and/or handled.

Run the code below to see how exception handling works:

```
<?php
function distance($speed, $time){

    if($time <= 0){
        throw new Exception('Time cannot be zero or negative.');// Throw exception if time is neg
    } else{

        $d = $speed*$time;
        echo "$speed * $time = $d";
    }
}
```

```

try{
    distance(10,2);
    distance(30,-4); //code will stop execution at this point (due to negative time) and start finally block
    distance(15,3);

    echo 'All calculations done!'; // If an exception is thrown, this line will not execute
}

// catch block is executed when an exception is thrown in the try block
// an object $e of Exception class is created
catch(Exception $e){

    echo "\n". "Caught exception: " . $e->getMessage(); //Exception handling
}

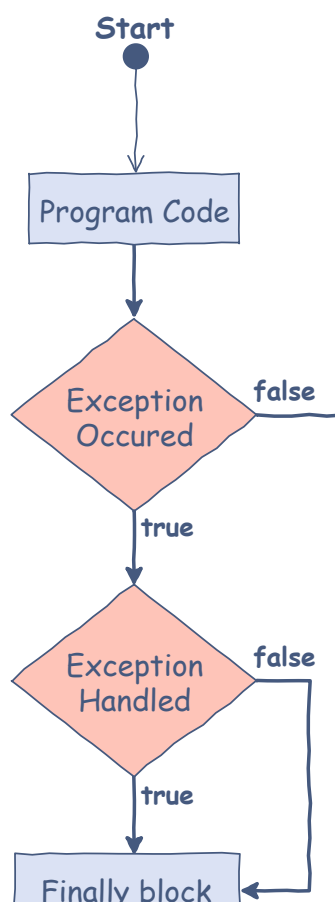
echo "\n"."Hello World!"; // Continue execution
?>

```



Note: The code will **terminate** if there is **no** catch for a thrown exception. So if you want the script to continue executing beyond the point where the exception has occurred, you must have at least **one** corresponding catch block for each try block.

The following figure illustrates the flow of a program with exception handling:



is executed

Methods

PHP's `Exception` class also provides the following methods for detailed information:

- `getCode()`
- `getFile()`
- `getLine()`
- `getTraceAsString()`

Other methods and properties of the `Exception` class can be seen in the [PHP documentation](#).

php without_catch

php with_catch

```
<?php
function division($a, $b){

    if($b ==0){
        throw new Exception('Divisor is zero'); // Throw exception if divisor is zero
    } else{

        $c = $a/$b;
        echo "$a / $b = $c";
    }
}

try{
    division(10, 2);
    division(15, 0);
    division(30, -4);

    echo 'All calculations done!'; // If an exception is thrown, this line will not execute
}

// write your catch statement here

echo "\n"."Hello World!"; // Continue execution
?>
```



The code above does not work since there is no catch statement in it.

Can you write a catch statement of your own on line 22 in the code and see if it runs? If you can't write it just yet, don't fret: see the code tab on the right.

to finish. If you can't write it just yet, don't fret, see the code that's on the right.

In the next lesson, we will learn how custom exceptions can be used in PHP.