PHP Exception Handling

Topics : PHP
Written on March 09, 2021

What is an Exception in php?

An exception is an event that occurs during the execution of a program that interrupts the normal flow of execution of the application.
An exception is different from an error because an exception can be handled by the program itself whereas an error cannot be handled by the program itself.
There are many Exception as below.

1. IOException
2. ArithmeticException - It is called as divide by zero exception
3. SQLException
4. ClassNotFoundException
5. InvalidArgumentException
6. RuntimeException
7. OverflowException
8. OutOfBoundsException and more...

Exception class is super class of all above mentioned subclass Exception.

We can extend Exception class using extends keyword as like Inheritance.

Exception Handling in php

- Exception Handling is needed to prevent stopping execution of a program when runtime errors occurs in code.
- Exception handling is required when an exception interrupts the normal execution of the program or application.
- Exception Handling continue the execution of the code from another location in the code.

We can handle Exception using below keywords.

1. **try- catch block :** Try block will throw exception occurs in program. This exception will be caught and handled in catch block. Try block must have at least one corresponding catch block or finally block. Also Try block can have more than 1 catch block. Multiple catch blocks can be used to catch different classes of exceptions.
2. **throw :** Throw keyword used to throw an exception in program. throw must have at least one corresponding catch block. Also we can throw user defined exception in program using throws keyword**.**

3. **finally :** Always the finally block will be executed after the try and catch blocks regardless if an exception occurred or not.

PHP Syntax:
Below php syntax contains try block, multiple catch block and finally block.

getMessage() : Returns a exception description.

```php
<?php
try {
   // code to be executed.
   // Try block throws Exception if occured.
}catch(Exception $e) {
   // Exception can be caught and handled by Catch Block.
   echo $e->getMessage();
}catch(InvalidArgumentException $e) {
   //multiple catch blocks within try catch
   echo $e->getMessage();
}finally {
   echo "This Code is always executed!";
}
?>
```

Below php syntax contains try block, catch block and throw keyword.

```php
<?php
  try {
    throw new Exception($error);
    // Below code is not executed.
    echo 'Never executed!';
  }catch(Exception $e) {
    echo 'Caught exception: '.$e->getMessage();
  }
  //Execution will be Continued
  echo 'Continue execution due to exception handling';
?>
```

PHP Example 1:

```php
<?php
//code by aryatechno
class Aryatechno extends Exception {

   function learn($dividend, $divisor){
     try {
        // code to be executed.
        // Try block throws Exception if occured.
        echo "</br>Inside try block";


        if ($divisor == 0) {
          throw new Exception("Divide by Zero");
```

```php
      }else
      {
         $num=  $dividend/$divisor;
      }
    echo "</br>Code after exception can not be executed.";
   }catch(Exception $e) {
      // Exception can be caught and handled by Catch Block.
      echo "</br> Exception Error : ".$e->getMessage();
   }catch(InvalidArgumentException $e) {
      //multiple catch blocks within try catch
      echo "</br>Invalid Argument Exception : ".$e->getMessage();
   }finally {
      echo "</br> This Code is always executed in finally block!";
   }
  }
}

$obj = new Aryatechno;
$obj->learn(12,6);
$obj->learn(14,0);
?>
```

Output :

Inside try block
Code after exception can not be executed.
This Code is always executed in finally block!
Inside try block
Exception Error : Divide by Zero
This Code is always executed in finally block!

Output :