Date:2021-02-04

* Throwing an Exception:
  + Exception handling are mainly used to handle errors in program.
  + Ex:
    - * <?php
      * function divide($num1,$num2){
      * if ($num2 == 0) {
      * throw new Exception("Cant divide by zero..");
      * } else {
      * return $num1 / $num2 ;
      * }
      * }
      * divide(5,0);
      * ?>
* try, throw, and catch:
  + it is used for handling an error.
  + Ex:
    - * <?php
      * $age = 10;
      * try {
      * if ($age > 18) {
      * echo "Done...";
      * } else {
      * throw new Exception("Your age is not enougth...");
      * }
      * } catch (Exception $e) {
      * echo "Error: ".$e->getMessage();
      * }
      * ?>
* Custom Exception:
  + We also set a custom exception in php.
  + Ex:
    - * <?php
      * $server = 'localhost';
      * $uname = 'root';
      * $pass = '';
      * $dbname = 'php\_test1';
      * class ServerException extends Exception {}
      * class DatabaseException extends Exception {}
      * try {
      * if (!@$conn = mysqli\_connect($localhost,$uname,$pass)) {
      * throw new ServerException("Could not connect with server..");
      * } elseif (!@mysqli\_select\_db($dbname)) {
      * throw new DatabaseException("Cant select Database...");
      * } else {
      * echo "Connected.........";
      * }
      * } catch(ServerException $e) {
      * echo "Error : ".$e->getMessage();
      * } catch(DatabaseException $e) {
      * echo "Error : ".$e->getMessage();
      * }
      * ?>
* Custom error message:
  + We can also set a custom error message in Exception class
  + Ex:
    - * <?php
      * $server = 'localhost';
      * $uname = 'root';
      * $pass = '';
      * $dbname = 'php\_test1';
      * class ServerException extends Exception {
      * public function getErrorMessage(){
      * return "Error Caugth on ".$this->getLine()."in File ".$this->getFile()."Code is :".$this->getCode();
      * }
      * }
      * class DatabaseException extends Exception {
      * public function getErrorMessage(){
      * return "Error Caugth on ".$this->getLine()."in File ".$this->getFile()."Code is :".$this->getCode();
      * }
      * }
      * try {
      * if (!@$conn = mysqli\_connect($localhost,$uname,$pass)) {
      * throw new ServerException("Could not connect with server..",405);
      * } elseif (!@mysqli\_select\_db($conn,$dbname)) {
      * throw new DatabaseException("Cant select Database...",101);
      * } else {
      * echo "Connected.........";
      * }
      * } catch(ServerException $e) {
      * echo "Error : ".$e->getErrorMessage();
      * } catch(DatabaseException $e) {
      * echo "Error : ".$e->getErrorMessage();
      * }
      * ?>
* Class create properties methods and change property values:
  + Ex:
    - * <?php
      * /\*\*
      * \* This class is for Manage a Bank Detail
      * \*/
      * class BankAccount
      * {
      * //create a property named balance.
      * public $currentBalance = 0;
      * //method for get current balance
      * function currentBalance(){
      * echo $this->currentBalance;
      * }
      * //method for depositr a money
      * function deposite($amt){
      * //we cheange the value of property currentBalance
      * $this->currentBalance = $this->currentBalance + $amt;
      * }
      * //method for withdraw a money
      * function withdraw($amt){
      * if ($this->currentBalance < $amt) {
      * echo "<br>No sufficiant Balance...";
      * } else {
      * $this->currentBalance = $this->currentBalance - $amt;
      * }
      * }
      * }
      * //create a instance of class
      * $jay = new BankAccount();
      * $jay->currentBalance();
      * $jay->deposite(500);
      * $jay->currentBalance();
      * $jay->withdraw(200);
      * $jay->currentBalance();
      * ?>
* Public, private and protected:
  + Public: scope to make that property/method available from anywhere, other classes and instances of the object.
  + Private: scope when you want your property/method to be visible in its own class only.
  + Protected : scope when you want to make your property/method visible in all classes that extend current class including the parent class.
* Constant:
  + For create a constant value we use const keyword.
  + For access a constant outside class we must use a classname::constant variable name.
  + For constant no need to prefix a $ for variable name.
  + For access constant inside class or in any method of class we use self::const\_var\_name.
  + Ex:
    - * <?php
      * /\*\*
      * \* class for practice a constant variable
      * \*/
      * class Circle
      * {
      * const pi = 3.14;
      * function area($radious){
      * return self::pi \* ($radious \* $radious);
      * }
      * }
      * $areas = new Circle();
      * echo $areas->area(10);
      * //for access constant outside class
      * echo $areas::pi;
      * echo Circle::pi;
      * ?>
* Class constructor:
  + Constructor call automatically when instance is created.
  + Ex:
    - * <?php
      * /\*\*
      * \* class for test a constractor
      * \*/
      * class test
      * {
      * // function \_\_construct()
      * // {
      * // echo "This is without parameter...";
      * // }
      * function \_\_construct($data)
      * {
      * echo "This is with parameter... and data is $data";
      * }
      * }
      * //call automatically when instance is created....
      * //$test1 = new test();
      * new test('hello');
      * ?>
* Multiple instance:
  + We can also create a multiple instance of same classes.
  + And value of both instance are different.
  + So it is easy to use and we can use single code for multiple users…
* Extends:
  + It is used for inheritance a new class to a existing class.
  + Ex:
    - * <?php
      * /\*\*
      * \* Class for student general info.
      * \*/
      * class Student
      * {
      * public $name = '';
      * public $age = '';
      * public $gender = '';
      * public function getInfo($name,$age,$gender){
      * echo "Name is : $name.<br>";
      * echo "Age is : $age<br>";
      * echo "Gender is : $gender<br>";
      * }
      * }
      * /\*\*
      * \* class for institute info.
      * \*/
      * class Institute extends Student
      * {
      * public $class = "";
      * public $cgpa = "";
      * public function getAllInfo($name,$age,$gender,$class,$cgpa){
      * $this->getInfo($name,$age,$gender);
      * echo "Class is : $class.<br>";
      * echo "CGPA is : $cgpa.";
      * }
      * }
      * //creating a instance of a institute class because there are no need to create instance of student class becoz it is inherited
      * $data = new Institute();
      * $data->getAllInfo('jay','21','male','MCA',8.57);
      * ?>