

PHP

- Destructor → In, destructor are called when a explicitly destroy an object or when all the references to the object go out of scope
- ↳ destructor have special name

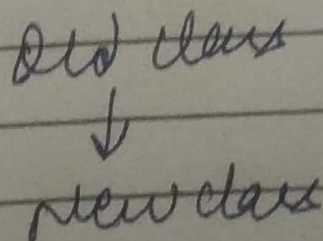
PHP -- destruct

- ↳ destructor don't have any arguments
- ↳ Syntax →

```
function __destruct ()  
{  
    echo "destroyed";  
}
```

Inheritance →

- ① The mechanism of deriving a new class from old one is called inheritance or derivation



Old class → Super class, Base class,
Parent class

Child class → new class, subclass,
Extended class

Parent class
↓
Child class

Father
• name
• money
• business

Son

• Job
• Access property
of father

Class A

```
{  
public & id  
public & name  
}
```

Class B extends A

```
{  
public & salary  
}
```

};

Declaration of Child
class

class child class extends parent
name & class name

```
{  
members of child class  
}
```

};

Type of Inheritance

① Single Inheritance → Parent class
↓
Child class

② Multiple ×

③ Multi-level → Father

④ Hierarchical

↓
Son

↓
Grandson

Single

Class Parent

→ Father

↓
son

↓
daughter

↓
son

Class Child extends Parent

Multi-level

Class Father

↳

Class son extends Father

↳

Class grandson extends son

↳

Hierarchical

```
Class Father  
{  
  4;  
}  
Class Son extends Father  
{  
  4;  
}  
Class Daughter extends Father  
{  
  4;  
}
```

~~Basic~~ Abstract class \Rightarrow is declared
with abstract ~~method~~ keywords.
 \hookrightarrow It can have abstract or non
abstract methods.
 \hookrightarrow Object of abstract class can't be
created

Abstract method \Rightarrow
A method that is declared as abstract
& doesn't have implementation
abstract function disp(); // no body.

abstract class father → Object of abstract class can't be created

```

    {
        abstract function disp();
    }

```

Class Son extends Father

```

    {
        public function disp()
        {
            echo "Abstract defined";
        }
    }

```

```

<?PHP
class Father {
    function disp() {
        echo "Method";
    }
}

$obj = new Father;
$obj → disp();
??

```

```

<?PHP
abstract class Father {
    function disp() {
        echo "Method";
    }
    abstract function ab();
}

class Son extends Father {
    public function ab() {
        echo "Abstract";
    }
}

$obj = new Obj;
$obj → ab();
??

```

If we use abstract class then shows error
 ∴ we can't create the obj of abstract class

PHP → Abstract

abstract method → mandatory
class to make class
abstract

final →

final keyword is used to create final
method or class

→ final method can't be overridden in
child class

→ final class can't be inherited it means
we can't create subclass of final class

class A {

final function disp() {
 echo "super class";
}

class B extends A {

function disp() {
 echo "child class";
} }
??

→ error → can't override
final method.

final class →

final class A {

function disp();
}

class B extends A

→ error → class B may not inherit
from final class A
→ ∴ we can't
create ∴ class
is final