# The City Lit Institute

##### Department of Computing

## Keeley Street, Holborn, London WC2B 4BA

**OO PHP**

**(with Apache server)**

**LECTURER**: **ALEXANDER ADU- SARKODIE**

MSc. Telematics (IT & Telecom), MSc. Eng., Dip. Russ. Lang., Teach. Cert, AMIAEng (UK), MBCS (UK), MIfL(UK)

**Blog:** [**http://www.blogger.com/profile/14800490193632788559**](http://www.blogger.com/profile/14800490193632788559)

**iStudio:** [**http://www.goldhawk-college.com/istudio/**](http://www.goldhawk-college.com/istudio/)

Inheritance

**As applied to PHP, class inheritance is accomplished by using the extends keyword.**

**A class that inherits from another class is known as child class, or a sub-class. The class from which the child class inherits is known as the parent, or base class.**

**Example 1**

**<?php**

**//Define a base Employee class**

**class Employee {**

**private $name;**

**//Define a setter for the private $name member.**

**function setName($name){**

**if ($name == "") {**

**echo "Name cannot be blank!";**

**}**

**else $this -> name = $name;**

**}**

**//Define a getter for the private $name member to return the value**

**function getName() {**

**return "My name is ".$this -> name." <br />";**

**}**

**}//End of Employee class**

**//Now define an Executive class that inherits from Employee**

**class Executive extends Employee {**

**//Define a method unique to Executive**

**function pillageCompany() {**

**echo "I am selling the company assets to finance my yatch";**

**}**

**}//End Executive class**

**//Create a new executive object**

**$exec = new Executive();**

**//Call the setName() method defined in the Employee class**

**$exec -> setName ("Alexander");**

**//Call the Employee getName () method on the instantiated object.**

**echo $exec -> getName ();**

**//Now call the local method defined with the extended class, pillageCompany method for association**

**$exec -> pillageCompany ();**

**?>.**

**Example 2**

**<?php**

**//Define a base Employee class**

**class Employee {**

**private $name;**

**//Define a setter for the private $name member.**

**function setName ($name){**

**if ($name == "") {**

**echo "Name cannot be blank!";**

**}**

**else $this -> name = $name;**

**}**

**//Define a getter for the private $name member to return the value**

**function getName () {**

**return "My name is ".$this -> name." <br />";**

**}**

**}//End of Employee class**

**//1 Now define an Executive class that inherits from Employee**

**class Executive extends Employee {**

**//Define a method unique to Employee**

**function pillageCompany () {**

**echo "I am selling company assets to finance my yatch.";**

**}**

**}//End Executive class**

**//Create a new executive object**

**$exec = new Executive();**

**//Call the setName () method defined in the Employee class**

**$exec -> setName ("Alexander");**

**//Call the Employee getName () method on the $exec object**

**echo $exec -> getName();**

**//2. Extending Executive: class CEO extends Executive**

**class CEO extends Executive {**

**function getFaceLift () {**

**echo " I need some square jaws";**

**}**

**}**

**$ceo = new CEO();**

**$ceo -> setName ("Marvin");**

**$ceo -> getName ();**

**//Now call the pillageCompany method for association**

**$ceo -> pillageCompany();**

**//Now call the getFaceLift in-built method for ceo**

**$ceo ->getFaceLift ();**

**?>**