Encapsulation, Inheritance, Polymorphism, Abstraction

**What is Object Oriented Programming and System (OOPS)?**

The Object Oriented Programming (OOP) is a programming model where programs are organized around object and data rather than action and logic.  
Object Oriented Programming (OOP) allow decomposition of a problem into a number of entities called object and then builds data and function around these objects.

1. The program is divided into number of small units called object. The data and function are build around these objects.  
2. The data of the objects can be accessed only by the functions associated with that object.  
3. The functions of one object can access the functions of other object.

All the programming languages supporting Object Oriented Programming (OOP) will be supporting these three main concepts:

1- Encapsulation  
2- Inheritance  
3- Polymorphism

**What is encapsulation in Object Oriented Programming (OOP)?**

Wrapping up data member and method together into a single unit (i.e. Class) is called Encapsulation.  
Encapsulation is like enclosing in a capsule. That is enclosing the related operations and data related to an object into that object.  
Encapsulation is like your bag in which you can keep your pen, book etc. It means this is the property of encapsulating members and functions.

1. class YourBag
2. {
3. book(); *// calling function of book*
4. pen(); *// calling function of pen*
5. ReadBook(); *// calling function readBook*
6. }

Encapsulation means hiding the internal details of an object, i.e. how an object does something.  
Encapsulation prevents clients from seeing its inside view, where the behaviour of the abstraction is implemented.  
Encapsulation is a technique used to protect the information in an object from the other object.

Hide the data for security such as making the variables as private, and expose the property to access the private data which would be public.  
So, when you access the property you can validate the data and set it.

Example:  
<?php  
class YourMarks  
{  
private $mark;  
public Marks  
{  
get { return $mark; }  
set { if ($mark > 0) $mark = 10; else $mark = 0; }  
}  
}  
?>

class YourMarks

{  
 private $mark;  
 public Marks  
 {  
 get { return $mark; }  
 set { if ($mark > 0) $mark = 10; else $mark = 0; }  
 }  
 }

Below the real life example of encapsulation:

1. I am giving example of Mobile Phone and it’s Manufacture:

Let’s take example of Mobile Phone and Mobile Phone Manufacturer  
Suppose you are a Mobile Phone Manufacturer and you designed and developed a Mobile Phone design(class), now by using machinery you are manufacturing a Mobile Phone(object) for selling, when you sell your Mobile Phone the user only learn how to use the Mobile Phone but not that how this Mobile Phone works.

This means that you are creating the class with function and by making object (capsule) of it you are making availability of the functionality of you class by that object and without the interference in the original class.

2. I am giving an another example of real life (daily use) that is “TV operation”. Many people’s operate TV in daily life.

It is encapsulated with cover and we can operate with remote and no need to open TV and change the channel.  
Here everything is in private except remote so that anyone can access not to operate and change the things in TV.

What is inheritance in Object Oriented Programming (OOP)?

When a class acquire the property of another class is known as inheritance.

Inheritance is process of object re-usability.

For example, A Child acquire property of Parents.

1. *// creating class as ParentClass*
3. public class ParentClass{
4. public ParentClass(){
5. echo "This is the Parent Constructor.";
6. }
8. public print(){
9. echo "I'm a Parent Class.";
10. }
11. }
13. *// creating class ChildClass and etends ParentClass*
15. public class ChildClass extends ParentClass
16. {
17. public ChildClass()
18. {
19. echo "This is the Child Constructor.";
20. }
22. public static Main()
23. {
24. $child = new ChildClass();
26. $child-&gt;print();
27. }
28. }
30. Output:
31. This is the Parent Constructor.
32. This is the Child Constructor.
33. I'm a Parent Class.

What is polymorphism in Object Oriented Programming (OOP)?

- Polymorphism means one name many forms.  
- One function behaves different forms.  
- In other words, “Many forms of a single object is called Polymorphism.”

I am giving real life example of Polymorphism:  
Example 1: Teacher and Students

- A Teacher behaves to student.  
- A Teacher behaves to his/her seniors.  
- Here teacher is an object but attitude is different in different situation.

What is inheritance in Object Oriented Programming (OOP)?

When a class acquire the property of another class is known as inheritance.  
Inheritance is process of object reusability.  
For example, A Child acquire property of Parents.

1. *// creating class as ParentClass*
3. public class ParentClass{
4. public ParentClass(){
5. echo "This is the Parent Constructor.";
6. }
8. public print(){
9. echo "I'm a Parent Class.";
10. }
11. }
13. *// creating class ChildClass and etends ParentClass*
15. public class ChildClass extends ParentClass
16. {
17. public ChildClass()
18. {
19. echo "This is the Child Constructor.";
20. }
22. public static Main()
23. {
24. $child = new ChildClass();
26. $child-&gt;print();
27. }
28. }
30. Output:
31. This is the Parent Constructor.
32. This is the Child Constructor.
33. I'm a Parent Class.

What is polymorphism in Object Oriented Programming (OOP)?

Polymorphism means one name many forms.  
One function behaves different forms.  
In other words, “Many forms of a single object is called Polymorphism.”

I am giving real life example of Polymorphism:

Example 1: Teacher and Students  
A Teacher behaves to student.  
A Teacher behaves to his/her seniors.  
Here teacher is an object but attitude is different in different situation.

Example 2: Person, Son and Employee  
Person behaves SON in house at the same time that person behaves EMPLOYEE in office.

Example 3: Mobile Phone  
Your mobile phone, one name but many forms

1. As phone  
2. As camera  
3. As mp3 player  
4. As radio

What is class in Object Oriented Programming (OOP)?

Class contains methods and properties.  
Class is a blueprint of an object that contains variables for storing data and functions to performing operations on these data.  
Class will not occupy any memory space and hence it is only logical representation of data.

1. class Student
2. {
3. public $age = 25;
4. function makrs(){
5. return '400';
6. }
7. }
9. 1. Student is a class name
10. 2. $age is a properties of student class
11. 3. marks is a method of student class

What is object in Object Oriented Programming (OOP)?

Object is an instance of a class.  
Object is a blue print of a class.  
Objects are the basic run-time entities in an object oriented system.They may represent a person,a place or any item that the program has to handle.  
Class will not occupy any memory space. Hence to work with the data represented by the class you must create a variable for the class, which is called as an object.

When an object is created by using the keyword new, then memory will be allocated for the class in heap memory area, which is called as an instance and its starting address will be stored in the object in stack memory area.

When an object is created without the keyword new, then memory will not be allocated in heap I.e. instance will not be created and object in the stack contains the value null.

When an object contains null, then it is not possible to access the members of the class using that object.

1. class Employee
2. {
3. *// code of class*
4. }
6. Syntax to create an object of class Employee:
7. $objEmp = new Employee();

What is abstraction in Object Oriented Programming (OOP)?

Abstraction provides you a generalized view of your classes or object by providing relevant information.  
Abstraction is the process of hiding the working style of an object, and showing the information of an object in understandable manner.  
I am giving real life Example of Abstraction:

Suppose you have an object Mobile Phone.  
Suppose you have 3 mobile phones as following

Nokia 1400 (Features- Calling, SMS)  
Nokia 2700 (Features- Calling, SMS, FM Radio, MP3, Camera)  
Black Berry (Features- Calling, SMS, FM Radio, MP3, Camera, Video Recording, Reading E-mails)

Abstract information (Necessary and Common Information) for the object “Mobile Phone” is make a call to any number and can send SMS.”  
so that, for mobile phone object you will have abstract class like following:-

1. abstract class MobilePhone
2. {
3. public Calling(); *// calling function*
4. public SendSMS(); *// calling function*
5. }
7. public class Nokia1400 extends MobilePhone
8. {
9. *// code here for Nokia 1400 class*
10. }
12. public class Nokia2700 extends MobilePhone
13. {
14. public FMRadio(); *// calling function*
15. public MP3(); *// calling function*
16. public Camera(); *// calling function*
17. }
19. public class BlackBerry extends MobilePhone
20. {
21. public FMRadio(); *// calling function*
22. public MP3(); *// calling function*
23. public Camera(); *// calling function*
24. public Recording(); *// calling function*
25. public ReadAndSendEmails(); *// calling function*
27. }

Abstraction means putting all the variables and methods in a class which are necessary.  
For example: – Abstract class and abstract method.  
Abstraction is the common thing.

Example:  
If somebody in your collage tell you to fill application form, you will fill your details like name, address, data of birth, which semester, percentage you have got etc.

If some doctor gives you an application to fill the details, you will fill the details like name, address, date of birth, blood group, height and weight.

See in the above example what is the common thing?  
Age, name, address so you can create the class which consist of common thing that is called abstract class.  
That class is not complete and it can inherit by other class.

Difference between Abstraction and Encapsulation :-  
Abstraction

1. Abstraction solves the problem in the design level.  
2. Abstraction is used for hiding the unwanted data and giving relevant data.  
3. Abstraction lets you focus on what the object does instead of how it does it  
4. Abstraction - Outer layout, used in terms of design.  
For Example:-  
Outer Look of a Mobile Phone, like it has a display screen and keypad buttons to dial a number.

Encapsulation

1. Encapsulation solves the problem in the implementation level.  
2. Encapsulation means hiding the code and data into a single unit to protect the data from outside world.  
3. Encapsulation means hiding the internal details or mechanics of how an object does something.  
4. Encapsulation - Inner layout, used in terms of implementation.  
For Example: - Inner Implementation detail of a Mobile Phone, how keypad button and Display Screen are connect with each other using circuits.

The easier way to understand Abstraction and encapsulation is as follows:-

Take an example of Mobile Phone:-

You have a Mobile Phone, you can dial a number using keypad buttons. Even you don’t know how these are working internally. This is called Abstraction. You have the only information that is needed to dial a number. But not its internal working of mobile.  
But how the Mobile Phone internally working?, how keypad buttons are connected with internal circuit? is called Encapsulation.