**Object** means a real word entity such as pen, chair, table etc.

**Object**

Object is the instance of a class.

an object is a self-contained entity that consists of both [data](http://www.webopedia.com/TERM/D/data.html) and [procedures](http://www.webopedia.com/TERM/P/procedure.html) to manipulate the data. Generally, any item that can be individually [selected](http://www.webopedia.com/TERM/S/select.html) and manipulated.

**Class:**

**Collection of objects** is called class.Class is a template for an object.

#### Abstraction

#### Abstraction means to show only necessary details to the client(user) of the object.

#### Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don't know the internal processing.

# Encapsulation

In programming, the process of combining elements to create a new entity. For example, a [procedure](http://www.webopedia.com/TERM/P/procedure.html) is a type of encapsulation because it combines a series of computer instructions.

[Object-oriented programming languages](http://www.webopedia.com/TERM/O/object_oriented_programming_OOP.html) rely heavily on encapsulation to create high-level [objects](http://www.webopedia.com/TERM/O/object.html).

**Difference between data encapsulation and data abstraction**

Data Encapsulation simply means wrapping relevant data together where as Data abstraction means abstracting out the the underneath logic and expose only the relevant part to the user.

|  |  |
| --- | --- |
| **Abstraction** | **Encapsulation** |
| 1. Abstraction solves the problem in the design level. | 1. Encapsulation solves the problem in the implementation level. |
| 2. Abstraction is used for hiding the unwanted data and giving relevant data. | 2. Encapsulation means hiding the code and data into a single unit to protect the data from outside world. |
| 3. Abstraction lets you focus on what the object does instead of how it does it | 3. Encapsulation means hiding the internal details or mechanics of how an object does something. |
| 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. | 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. |

# Inheritance

Inheritance is the process by which one object acquires the properties of another object.

# polymorphism

polymorphism is a feature that allows one interface to be used for a general class of actions.

Basically, let’s say we had an apple, banana, and orange class. They’re all different, but they all fall under the same category as fruit. So, it would make sense to create an interface named fruit and have all these classes implement it. That’s what polymorphism is. It’s a bunch of classes with different functionality and structure, but all use the same interface.

**Arrays**

In PHP, there are three types of arrays:

* **Indexed arrays** - Arrays with a numeric index
* **Associative arrays** - Arrays with named keys
* **Multidimensional arrays** - Arrays containing one or more arrays

# Constructor

#### This is a method you can use to set initial values for field variables. When the object is created, Java calls the constructor first. Any code you have in your constructor will then get executed. You don't need to make any special calls to a constructor method - they happen automatically when you create a new object.

#### Constructor have no return type.

# Recursion

**A method that calls itself is said to be recursive.**

**Computation of the factorial of a number is the best example of recursion.**

**Public**

Public method or variable can be accessible from anywher. I mean from inside the class, out side the class and in child class also.

**Private**

Method or property with private visibility can only be accessible inside the class. You can not access private method or variable from outside of your class.

**Protected**

An attribute that is given a **protected** scope can only be accessed within the class and to any other classes that extend the base class.

Method or variable with protected visibility can only be access in the derived class. Or in other word in child class. Protected will be used in the process of inheritance.

**the main difference between protected and private class properties is that protected properties can also be accessed by extended classes.**

# Final Keyword

The **final keyword** in java is used to restrict the user. The java final keyword can be used in many context. Final can be:

1. variable
2. method
3. class

If you make any variable as final, you cannot change the value of final variable(It will be constant).

If you make any method as final, you cannot override it.

If you make any class as final, you cannot extend it.

### Q) Is final method inherited?

Ans) Yes, final method is inherited but you cannot override it.

### Q) What is blank or uninitialized final variable?

A final variable that is not initialized at the time of declaration is known as blank final variable.

If you want to create a variable that is initialized at the time of creating object and once initialized may not be changed, it is useful. For example PAN CARD number of an employee.

It can be initialized only in constructor.

### Qu) Can we initialize blank final variable

Yes, but only in constructor.

### static blank final variable

A static final variable that is not initialized at the time of declaration is known as static blank final variable. It can be initialized only in static block.

### Q) Can we declare a constructor final?

No, because constructor is never inherited.

### Why use inheritance in java

* For Method Overriding.
* For Code Reusability.

# super keyword

The **super** keyword in java is a reference variable that is used to refer immediate parent class object.

Whenever you create the instance of subclass, an instance of parent class is created implicitly i.e. referred by super reference variable.

# Abstract class

A class that is declared with abstract keyword, is known as abstract class in java. It can have abstract and non-abstract methods

A class that is declared as abstract is known as **abstract class**. It needs to be extended and its method implemented. It cannot be instantiated.

# Packages

Package is both a naming and visibility control mechanism. You can define classes inside a package that are not accessible by code outside that package

## Association

It represents a relationship between two or more objects where all objects have their own lifecycle and there is no owner.

**Example**: Manager uses a swipe card to enter XYZ premises.

In this example, the manager object and the swipe card object use each other but they have their own object life time. In other words, they can exist without each other. The most important point in this relationship is that there is no single owner.

## Aggregation

It is a specialized form of Association where all object have their own lifecycle but there is ownership.

**Example:** Manager has workers who work under him.

It denotes the same type of relationship like association but with a difference that one of them is an owner.

the Worker object can have its own life time which is completely disconnected from the Manager object. Looking from a different perspective, it means that if the Manager object is deleted, the Worker object does not die.

## Composition

In this relationship child objects does not have their lifecycle without Parent object. If a parent object is deleted, all its child objects will also be deleted. This represents “death” relationship.

**Example:**

1. Manager has the responsibility of ensuring that the project is successful.
2. Manager's salary will be judged based on project success.

Below is the conclusion from analyzing the above requirements:

1. Manager and the project objects are dependent on each other.
2. The lifetimes of both the objects are the same. In other words, the project will not be successful if the manager is not good, and the manager will not get good increments if the project has issues.

**Data Structures**

Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way.

In simple language, Data Structures are structures programmed to store ordered data, so that various operations can be performed on it easily.

**CURL**

|  |
| --- |
|  |
| cURL is a way you can hit a URL from your code to get a html response from it. cURL means client URL which allows you to connect with other URLs and use their responses in your code.  **Composer** |

Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

Suppose:

a) You have a project that depends on a number of libraries.

b) Some of those libraries depend on other libraries.

Composer:

c) Enables you to declare the libraries you depend on.

d) Finds out which versions of which packages can and need to be installed, and installs them (meaning it downloads them into your project).

**Static Variables**

Static methods and properties in php can directly accessible without creating object of class.

Static properties of class is a property which is directly accessible from class with the help of **::**(scope resolution operator).

Normally, when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. To do this, use the **static** keyword when you first declare the variable then, each time the function is called, that variable will still have the information it contained from the last time the function was called.

**WORDPRESS**

# What encryption method does Wordpress use to store its passwords?

**Ans: Portable PHP password hashing framework (PH Pass)**

By default PHPass is used to generate password hashes

Template Hierarchy

<http://cdn.tutsplus.com/wp/uploads/legacy/090_WPCheatSheets/WP_CheatSheet_TemplateMap.jpg>