**OOPS Interview Questions**

1) Explain the rationale behind Object Oriented concepts?  
Object oriented concepts form the base of all modern programming languages. Understanding the basic concepts of object-orientation helps a developer to use various modern day programming languages, more effectively.

2) Explain about Object oriented programming?  
Object oriented programming is one of the most popular methodologies in software development. It offers a powerful model for creating computer programs. It speeds the program development process, improves maintenance and enhances reusability of programs.

**3) Explain what is an object?**  
An object is a combination of messages and data. Objects can receive and send messages and use messages to interact with each other. The messages contain information that is to be passed to the recipient object.  
  
4) Explain the implementation phase with respect to OOP?  
The design phase is followed by OOP, which is the implementation phase. OOP provides specifications for writing programs in a programming language. During the implementation phase, programming is done as per the requirements gathered during the analysis and design phases.

5) Explain about the Design Phase?  
In the design phase, the developers of the system document their understanding of the system. Design generates the blue print of the system that is to be implemented. The first step in creating an object oriented design is the identification of classes and their relationships.

6) Explain about a class?  
Class describes the nature of a particular thing. Structure and modularity is provided by a Class in object oriented programming environment. Characteristics of the class should be understandable by an ordinary non programmer and it should also convey the meaning of the problem statement to him. Class acts like a blue print.

7) Explain about instance in object oriented programming?  
Every class and an object have an instance. Instance of a particular object is created at runtime. Values defined for a particular object define its State. Instance of an object explains the relation ship between different elements.

8) Explain about inheritance?  
Inheritance revolves around the concept of inheriting knowledge and class attributes from the parent class. In general sense a sub class tries to acquire characteristics from a parent class and they can also have their own characteristics. Inheritance forms an important concept in object oriented programming.  
9) Explain about multiple inheritance?  
Inheritance involves inheriting characteristics from its parents also they can have their own characteristics. In multiple inheritance a class can have characteristics from multiple parents or classes. A sub class can have characteristics from multiple parents and still can have its own characteristics.

10) Explain about encapsulation?  
Encapsulation passes the message without revealing (knowing) the exact functional details of the class. It allows only the relevant information to the user without revealing the functional mechanism through which a particular class had functioned.

11) Explain about abstraction?  
Abstraction simplifies a complex problem to a simpler problem by specifying and modeling the class to the relevant problem scenario. It simplifies the problem by giving the class its specific class of inheritance. Composition also helps in solving the problem to an extent.

12) Explain the mechanism of composition?  
Composition helps to simplify a complex problem into an easier problem. It makes different classes and objects to interact with each other thus making the problem to be solved automatically. It interacts with the problem by making different classes and objects to send a message to each other.

13) Explain about polymorphism?  
Polymorphism helps a sub class to behave like a parent class. When an object belonging to different data types respond to methods which have a same name, the only condition being that those methods should perform different function

.  
14) Explain about overriding polymorphism?  
Overriding polymorphism is known to occur when a data type can perform different functions. For example an addition operator can perform different functions such as addition, float addition etc. Overriding polymorphism is generally used in complex projects where the use of a parameter is more.  
**15) Explain about object oriented databases?**  
Object oriented databases are very popular such as relational database management systems. Object oriented databases systems use specific structure through which they extract data and they combine the data for a specific output. These DBMS use object oriented languages to make the process easier.  
16) Explain about parametric polymorphism?  
Parametric polymorphism is supported by many object oriented languages and they are very important for object oriented techniques. In parametric polymorphism code is written without any specification for the type of data present. Hence it can be used any number of times.  
17) What are all the languages which support OOP?  
There are several programming languages which are implementing OOP because of its close proximity to solve real life problems. Languages such as Python, Ruby, Ruby on rails, Perl, PHP, Coldfusion, etc use OOP. Still many languages prefer to use DOM based languages due to the ease in coding.

---------

1) Explain what is object oriented programming language?  
Object oriented programming language allows concepts such as modularity, encapsulation, polymorphism and inheritance. Similar is credited to be the first object oriented language. Objects are said to be the most important part of object oriented language. Concept revolves around making simulation programs around an object.  
2) Name some languages which have object oriented language and characteristics?  
Some of the languages which have object oriented languages present in them are ABAP, ECMA Script, C++, Perl, LISP, C#, Tcl, VB, Ruby, Python, PHP, etc. Popularity of these languages has increased considerably as they can solve complex problems with ease.  
3) Explain about UML?  
UML or unified modeling language is regarded to implement complete specifications and features of object oriented language. Abstract design can be implemented in object oriented programming languages. It lacks implementation of polymorphism on message arguments which is a OOPs feature.  
4) Explain the meaning of object in object oriented programming?  
Languages which are called as object oriented almost implement everything in them as objects such as punctuations, characters, prototypes, classes, modules, blocks, etc. They were designed to facilitate and implement object oriented methods.  
5) Explain about message passing in object oriented programming?  
Message passing is a method by which an object sends data to another object or requests other object to invoke method. This is also known as interfacing. It acts like a messenger from one object to other object to convey specific instructions.  
6) State about Java and its relation to Object oriented programming?  
Java is widely used and its share is increasing considerably which is partly due to its close resemblance to object oriented languages such as C and C++. Code written in Java can be transported to many different platforms without changing it. It implements virtual machine.  
7) What are the problems faced by the developer using object oriented programming language?  
These are some of the problems faced by the developer using object oriented language they are: -  
1) Object oriented uses design patterns which can be referred to as anything in general.  
2) Repeatable solution to a problem can cause concern and disagreements and it is one of the major problems in software design.  
8) State some of the advantages of object oriented programming?  
Some of the advantages of object oriented programming are as follows: -  
1) A clear modular structure can be obtained which can be used as a prototype and it will not reveal the mechanism behind the design. It does have a clear interface.  
2) Ease of maintenance and modification to the existing objects can be done with ease.  
3) A good framework is provided which facilitates in creating rich GUI applications.  
9) Explain about inheritance in OOPS?  
Objects in one class can acquire properties of the objects in other classes by way of inheritance. Reusability which is a major factor is provided in object oriented programming which adds features to a class without modifying it. New class can be obtained from a class which is already present.  
10) Explain about the relationship between object oriented programming and databases?  
Object oriented programming and relational database programming are almost similar in software engineering. RDBMS will not store objects directly and that’s where object oriented programming comes into play. Object relational mapping is one such solution.  
11) Explain about a class in OOP?  
In Object oriented programming usage of class often occurs. A class defines the characteristics of an object and its behaviors. This defines the nature and functioning of a specified object to which it is assigned. Code for a class should be encapsulated.

**12) Explain the usage of encapsulation?**Encapsulation specifies the different classes which can use the members of an object. The main goal of encapsulation is to provide an interface to clients which decrease the dependency on those features and parts which are likely to change in future. This facilitates easy changes to the code and features.

13) Explain about abstraction?  
Abstraction can also be achieved through composition. It solves a complex problem by defining only those classes which are relevant to the problem and not involving the whole complex code into play.

**14) Explain what a method is?**  
A method will affect only a particular object to which it is specified. Methods are verbs meaning they define actions which a particular object will perform. It also defines various other characteristics of a particular object.

Or

The **functions** which are declared in a class are called **methods**. A class **method** is exactly similar to **PHP functions**. Declaring a **method** in a class is an easy task, use one of the keyword public, protected, or private followed by a **method** name. public : The **method** can be accessed from outside the class

15) Name the different Creational patterns in OO design?  
There are three patterns of design out of which Creational patterns play an important role the various patterns described underneath this are: -  
1) Factory pattern  
2) Single ton pattern  
3) Prototype pattern  
4) Abstract factory pattern  
5) Builder pattern

16) Explain about realistic modeling?  
As we live in a world of objects, it logically follows that the object oriented approach models the real world accurately. The object oriented approach allows you to identify entities as objects having attributes and behavior.

17) Explain about the analysis phase?  
The anlaysis or the object oriented analysis phase considers the system as a solution to a problem in its environment or domain. Developer concentrates on obtaining as much information as possi

|  |  |
| --- | --- |
| **Questions : 9** | **What is difference between Static and Non-Static fields of a class ?** |
| **Answers : 9** | Non-Static values are also called as instance variables. Each object of the class has its own  copy of Non-Static instance variables. So when a new object is created of the same class it  will have completely its own copy of instance variables. While Static values have only one  copy of instance variables and will be shared among all the objects of the class. |

#### Destructor :

* It is similar to constructor method.
* Destructor method which actually works upon destroying an object.
* You can explicitly create a destructor method by naming it \_\_destruct().
* It will be invoked automatically by PHP at the end of the execution of your codes.