**Interpreter Examples**

Computer languages have been developed with interpreting statements in mind. These are some of them:

**Python**

A very popular, free, object oriented interpreted language created in 1991. It is also modular which means people can add extra parts to it for others to use. You will find Python being used as the back-end processing language on web servers.

**BASIC**

The grand-daddy of interpreted computer languages. Created nearly fifty years ago (1963) as an easy-to-learn language. Its strength is that it is interactive - it checks your code as you type it in and its syntax is simple.

**Java**

This is the second type of interpreter. It compiles source code into bytecode which then runs on a virtual machine. Java is very popular as the language to use for coding the business-logic within a company's business application.

For example, on on-line mortgage application form on a Building Society web site would most likely be connected to a Java server that is working through all the complex business rules for providing a customer quote.

Differences between C and JAVA

1. JAVA is Object-Oriented while C is procedural. Different Paradigms
2. Java is an Interpreted language while C is a compiled language.
3. C is a low-level language while JAVA is a high-level language.
4. C uses the top-down **{sharp & smooth}** approach while JAVA uses the bottom-up **{on the rocks}** approach.
5. Pointer goes backstage in JAVA while C requires explicit handling of pointers.
6. The Behind-the-scenes Memory Management with JAVA & the User-Based Memory Management in C.
7. JAVA supports Method Overloading while C does not support overloading at all.
8. Unlike C, JAVA does not support Preprocessors, & does not really them.
9. The standard Input & Output Functions.
10. Exception Handling in JAVA and the errors & crashes in C.

The advantages of Java are:

1. Java is easy to learn

Java was designed to be easy to use and is therefore much easier to write, compile, debug, run and learn than other programming languages.

1. Java is object-oriented

This allows you to create modular maintainable applications and reusable code.

1. Java is platform-independent

One of the most significant advantages of Java is its ability to move easily from one system to another. The ability to run the same code on many different systems is crucial to www, and Java succeeds at this by being platform-independent at the source and almost binary levels.

1. Java is distributed

Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it. Writing network programs in Java is like sending and receiving data to and from a file.

1. Java is secure

Java considers security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.

1. Java is robust

Robust means reliability. Java puts a lot of emphasis on early checking for possible errors, as Java compilers are able to detect many problems that would first show up during execution time in other languages.

1. Java is multithreaded

Multithreaded is the capability for a program to perform several tasks simultaneously within a program. In Java, multithreaded programming has been smoothly integrated into it, while in other languages, operating system-specific procedures have to be called in order to enable multithreading.

<http://www.techflirt.com/tutorials/oop-in-php/index.html>

# PHP Singleton Pattern – OOP

# <http://phpadvocate.com/blog/2011/04/php-using-a-singleton-pattern-with-oop/>

php nettuts

<http://net.tutsplus.com/category/tutorials/php/>

# [PHP Interview Questions and Answers](http://rajeshstutorials.blogspot.com/p/php-interview-questions-and-answers.html)

<http://rajeshstutorials.blogspot.com/p/php-interview-questions-and-answers.html>

# Uniﬁed Modeling Language(UML)

UML stands for Uniﬁed Modeling Language.UML is used to manage large and complex systems.  
  
With UML you can:

* Manage project complexity.
* create database schema.
* Produce reports.

**Types of UML Diagrams:**

1. Class Diagrams
2. Package Diagrams
3. Object Diagrams
4. Use Case Diagrams
5. Sequence Diagrams
6. Collaboration Diagrams
7. State chart Diagrams
8. Activity Diagrams
9. Component Diagrams
10. Deployment Diagrams

A static constructor is used to initialize any static data, or to perform a particular action that needs performed once only. It is called automatically before the first instance is created or any static members are referenced.

* Static constructors have the following properties:
* A static constructor does not take access modifiers or have parameters.
* A static constructor is called automatically to initialize the class before the first instance is created or any static members are referenced.
* A static constructor cannot be called directly.
* The user has no control on when the static constructor is executed in the program

**Database Questions:**

1. What is the difference between primary key and foreign key?

* A **primary key** is a field or combination of fields that uniquely identify a record in a table, so that an individual record can be located without confusion.
* A **foreign key** (sometimes called a referencing key) is a key used to link two tables together. Typically you take the primary key field from one table and insert it into the other table where it becomes a foreign key (it remains a primary key in the original table

1. What is Normalization 1NF, 2NF and 3NF (one, second and third normal form)?

* Eliminating redundant data
  + 1NF: ميكونش فيه اي عمود مركب يعني مكون من عمودين وميكونش قيم مختلفه
  + 2NF: جميع الأعمدة العادية في الجدول تعتمد على المفتاح الأساسي كاملا وليس جزء منه
  + 3NF: ميكونش فيه عمود بيعتمد علي عمود تاني

1. Relationship types (one to one, one to many and many to many)?

* One-to-one: Both tables can have only one record on either side of the relationship. Each primary key value relates to only one (or no) record in the related table. They're like spouses—you may or may not be married, but if you are, both you and your spouse have only one spouse. Most one-to-one relationships are forced by business rules and don't flow naturally from the data. In the absence of such a rule, you can usually combine both tables into one table without breaking any normalization rules.
* One-to-many: The primary key table contains only one record that relates to none, one, or many records in the related table. This relationship is similar to the one between you and a parent. You have only one mother, but your mother may have several children.
* Many-to-many: Each record in both tables can relate to any number of records (or no records) in the other table. For instance, if you have several siblings, so do your siblings (have many siblings). Many-to-many relationships require a third table

1. What are the advantages and disadvantages of Databases?

**Advantages**

* Reduced data redundancy
* Reduced updating errors and increased consistency
* Greater data integrity and independence from applications programs
* Improved data access to users through use of host and query languages
* Improved data security
* Reduced data entry, storage, and retrieval costs
* Facilitated development of new applications program

**Disadvantages**

* Database systems are complex, difficult, and time-consuming to design
* Substantial hardware and software start-up costs
* Damage to database affects virtually all applications programs
* Extensive conversion costs in moving form a file-based system to a database system
* Initial training required for all programmers and users

1. Examples of Databases ( Oracle, Microsoft SQL Server, DB2, MySQL)
2. What is RDBMS (Relational database management system)?

* RDBMS stands for Relational Database Management System. RDBMS data is structured in database tables, fields and records. Each RDBMS table consists of database table rows. Each database table row consists of one or more database table fields.
* RDBMS store the data into collection of tables, which might be related by common fields (database table columns). RDBMS also provide relational operators to manipulate the data stored into the database tables. Most RDBMS use [SQL](http://www.sql-tutorial.net/) as database query language.

1. What is an Entity?

An Entity is a person, place, thing or concept about which data can be collected. Examples include EMPLOYEE, HOUSE, CAR

1. What is Weak Entity?

* A Strong Entity is one that exists on its own, independent of other entities.
* A Weak Entity is one whose existence depends on another entity.

1. What are the different types of attributes?

* Simple and Composite Attribute
* Single Valued and Multi Valued attribute
* Stored and Derived Attributes
* Complex Attribute

**SQL Questions:**

1. What is SQL (Structured Query language)?
2. What is the difference between Where and Having?

* ال where تستخدم مع الحقول العادية التي من قاعدة البيانات
* ام ال having فهي تستخدم مع الحقول التجميعية فهي تعمل نفس عمل ال Select ولكن على النتائج التي ظهرت من جملة ال Group By يعني أن وجودها مرهون بوجود ال Group By

1. What is the order of Select statement clauses (Select, From, Where)?
2. What is the different between Delete and Truncate?
   * Truncate: fast and cannot undo
   * Delete: can undo this delete
3. Having and Group By, Which one is mandatory or prerequisite?
4. Define View and its types?
5. Define Index and its types?
6. Define Subquery and its types?
7. What is Join and the different types of Join?
8. What is the difference between DML, DDL and DCL?
9. What is the difference between Group by and Order By?
10. What is the difference between Delete and Drop?
11. After creating Database table can you add additional column?
12. What is the difference between Function and Stored Procedure?
13. What are the different types of Constraints?