**Web Design and Development**

**Name: Amina Iftikhar Ahmed**

**Roll No:11**

**Assignment**

**Topic:**

Low level, High level Language And PHP

**Low Level Language:**

A low-level programming language is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that provides little or no [abstraction](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) from a computer's instruction set architecture commands or functions in the language map closely to processor instructions. Generally, this refers to either [machine code](https://en.wikipedia.org/wiki/Machine_code) or [assembly language](https://en.wikipedia.org/wiki/Assembly_language). The word **"low"** refers to the small or nonexistent amount of [abstraction](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) between the language and machine language; because of this, low-level languages are sometimes described as being **"close to the hardware"**. Programs written in low-level languages tend to be relatively [non-portable](https://en.wikipedia.org/wiki/Software_portability).

**Low-level languages are oriented toward exploiting hardware.**

Low-level languages can convert to machine code without a compiler or interpreter. S[econd-generation programming languages](https://en.wikipedia.org/wiki/Second-generation_programming_language) use a simpler processor called an [assembler](https://en.wikipedia.org/wiki/Assembly_language#Assemble) and the resulting code runs directly on the processor. A program written in a low-level language can be made to run very quickly, with a small [memory footprint](https://en.wikipedia.org/wiki/Memory_footprint). An equivalent program in a [high-level language](https://en.wikipedia.org/wiki/High-level_language) can be less efficient and use more memory. Low-level languages are simple, but considered difficult to use, due to numerous technical details that the programmer must remember. By comparison, a [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language) isolates execution semantics of a computer architecture from the specification of the program, which simplifies development.

Low-level programming languages are sometimes divided into two categories:

***first generation* and *second generation.***

**Low level programming languages:**

* Binary
* Machine
* Assembly
* Compiler
* BrainFked

**High Level Language:**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), a **high-level programming language** is a [programming language](https://en.wikipedia.org/wiki/Programming_language) with strong [abstraction](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) from the details of the [computer](https://en.wikipedia.org/wiki/Computer). In contrast to [low-level programming languages](https://en.wikipedia.org/wiki/Low-level_programming_language), it may use [natural language](https://en.wikipedia.org/wiki/Natural_language) *elements*, be easier to use, or may automate (or even hide entirely) significant areas of computing systems (e.g. [memory management](https://en.wikipedia.org/wiki/Memory_management)), making the process of developing a program simpler and more understandable than when using a lower-level language.

**High-level languages are oriented toward expressing ideas.**

In the 1960s, high-level programming languages using a [compiler](https://en.wikipedia.org/wiki/Compiler) were commonly called [**autocodes**](https://en.wikipedia.org/wiki/Autocode)**: C**[OBOL](https://en.wikipedia.org/wiki/COBOL) and [Fortran](https://en.wikipedia.org/wiki/Fortran).

The first high-level programming language designed for computers was [Plankalkül](https://en.wikipedia.org/wiki/Plankalk%C3%BCl), created by [Konrad Zuse](https://en.wikipedia.org/wiki/Konrad_Zuse). [World War II](https://en.wikipedia.org/wiki/World_War_II), aside from the language's influence on the "Superplan" language by [Heinz Rutishauser](https://en.wikipedia.org/wiki/Heinz_Rutishauser) and also to some degree [Algol](https://en.wikipedia.org/wiki/ALGOL). The first significantly widespread high-level language was [Fortran](https://en.wikipedia.org/wiki/Fortran), a machine-independent development of IBM's earlier [Autocode](https://en.wikipedia.org/wiki/Autocode) systems. [Algol](https://en.wikipedia.org/wiki/ALGOL), defined in 1958 and 1960 by committees of European and American computer scientists, introduced [recursion](https://en.wikipedia.org/wiki/Recursion) as well as [nested functions](https://en.wikipedia.org/wiki/Nested_functions) under [lexical scope](https://en.wikipedia.org/wiki/Lexical_scope). It was also the first language with a clear distinction between [value](https://en.wikipedia.org/wiki/Call_by_value) and [name-parameters](https://en.wikipedia.org/wiki/Call_by_name) and their corresponding [semantics](https://en.wikipedia.org/wiki/Semantics). [structured programming](https://en.wikipedia.org/wiki/Structured_programming) concepts, such as the **while-do** and **if-then-else** constructs and its [syntax](https://en.wikipedia.org/wiki/Syntax) was the first to be described in formal notation – "[Backus–Naur form](https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form)" (BNF). During roughly the same period, [Cobol](https://en.wikipedia.org/wiki/Cobol) introduced [records](https://en.wikipedia.org/wiki/Record_(computer_science)) (also called structs) and [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) introduced a fully general [lambda abstraction](https://en.wikipedia.org/wiki/Lambda_abstraction) in a programming language for the first time.

**High level programming languages:**

* C++
* C#
* Cobol
* Fortran
* Java
* JavaScript
* Objective C
* Pascal
* Perl
* PHP

# **Question:**

# **Why is PHP called “Hypertext Preprocessor”?**

**Answer:**

PHP was written in the C programming language by Rasmus Lerdorf in 1994 for use in monitoring his online resume and related personal information. For this reason, PHP originally stood for "**Personal Home Page**". Lerdorf combined PHP with his own Form Interpreter, releasing the combination publicly as **PHP/FI** (generally referred to as PHP 2.0) on June 8, 1995. Two programmers, Zeev Suraski and Andi Gutmans, rebuilt PHP's core, releasing the updated result as PHP/FI 2 in 1997. The acronym was formally changed to **PHP: HyperText Preprocessor**, at this time. (This is an example of a recursive acronym: where the acronym itself is in its own definition.)

PHP is a server-side scripting language which originally stood for Personal Home Page and was created by Rasmus Lerdorf in 1994.

It was never intended to be a fully blown programming language as Rasmus had simply wanted a set of tools to help him maintain his homepage.

As the language developed and became what it is today the acronym changed to be the recursive version we have now: PHP: **Hypertext Preprocessor.**

The Hypertext Preprocessor part refers to **HTML, Hyper Text Markup Language**, and the fact that it is run on the server and so can be used to process the HTML before it is sent to the browser and processed again.