Module 01: Introduction to PHP

**What is PHP**

What is PHP:

Support class- 6 days (without Sujday)

Live class – Friday & Sunday (09:15 PM)

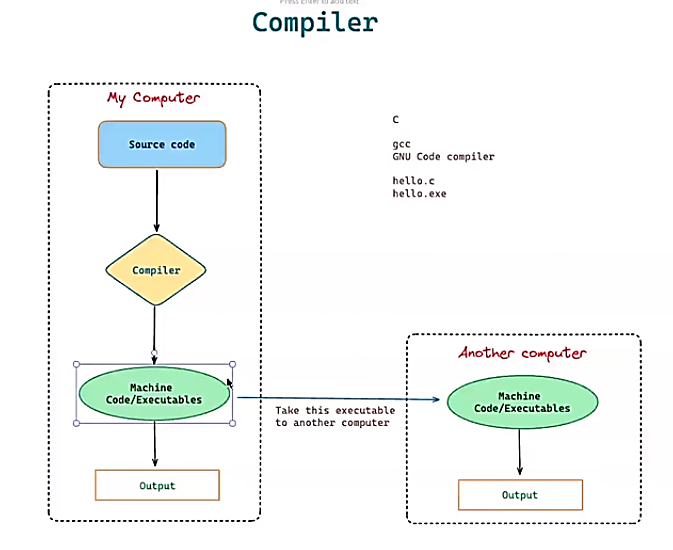
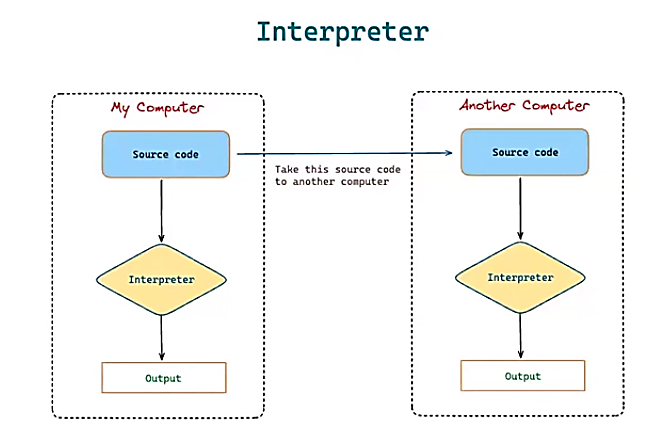
Task-assignment => deadline 2 week

Documentation - > Google docs / Notion.so

PHP creator- rasmus lerdorf (1995)

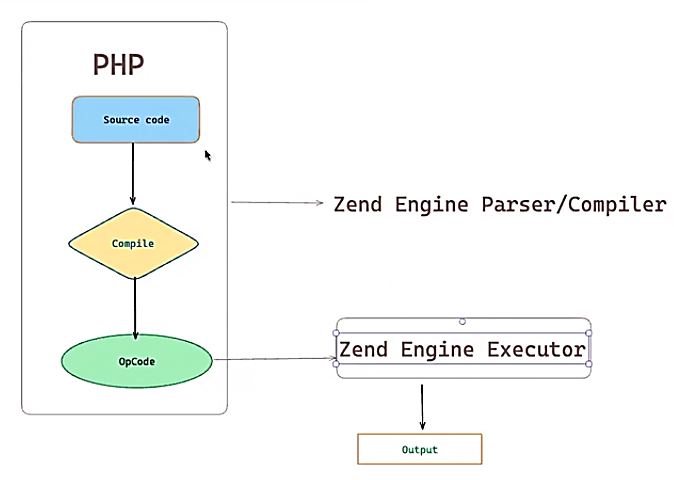
Php code execute through Zend Engine PHP এর 6 ভার্সনের রিলিজ শেষ পর্যন্ত হয়নি

OOP introduced in PHP v5

Php latest vesion 8.3

Compiled code is fast than interpreted code for sharing the mechine code

For interpreted code , every mechine needs an interpreter



পিএইচপি সোর্স কোড থেকে ফাইনাল কোডের প্রসেস তাই জেন্ড ইঞ্জিন করে

জেন্ড কোড C দিয়ে লেখা এবং পিএইচপি এর কোর হলো zend বিধায় পিএইচপি C দিয়ে লেখা বলা হয়

Go ল্যাংগুয়েজ এ সিনট্যাক্স অনেক কম বিধায় দ্রুত এবং সহজবোধ্য

**Compiler, Interpreter and How PHP code executes**

**Compiler and Interpreters**

* What compiling code means. <https://dev.to/arikaturika/code-compiling-explain-like-im-five-4mkj>
* Interpreted vs Compiled Programming Languages: What's the Difference? <https://www.freecodecamp.org/news/compiled-versus-interpreted-languages/>
* What's the Difference Between Interpreted Languages and Compiled Languages? <https://www.youtube.com/watch?v=KsZLPTRSleI>

**How PHP Interpreter works**

* <https://www.droptica.com/blog/how-php-interpreter-works/>
* <https://www.sitepoint.com/how-php-executes-from-source-code-to-render/>

**Understanding Unicode and Character sets**

* <https://medium.com/%E0%A6%AA%E0%A7%8D%E0%A6%B0%E0%A7%8B%E0%A6%97%E0%A7%8D%E0%A6%B0%E0%A6%BE%E0%A6%AE%E0%A6%BF%E0%A6%82-%E0%A6%AA%E0%A6%BE%E0%A6%A4%E0%A6%BE/encoding-encryption-hashing-%E0%A6%8F%E0%A6%95%E0%A6%87-%E0%A6%A8%E0%A6%BE%E0%A6%95%E0%A6%BF-%E0%A6%AD%E0%A6%BF%E0%A6%A8%E0%A7%8D%E0%A6%A8-f9d3f2650cb7>
* <https://ahmedshamim.com/posts/what-every-programmer-should-know-about-string>
* <https://www.smashingmagazine.com/2012/06/all-about-unicode-utf8-character-sets/>
* [https://web.archive.org/web/20240527070658/https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/](https://web.archive.org/web/20240527070658/https:/www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/)

**PHP Problem solving and DSA**

**Edge case** : যখন কোন একটা কেস মিস করা হয় যার কারণে ইরোর দেখায়

**Off by One error**: লুপ চালানোর সময় ইনডেক্সিং ভুলের কারণে একটা লুপ বেশি বা কম করে ফেলা

php তে native Data Structure মানে হলো array

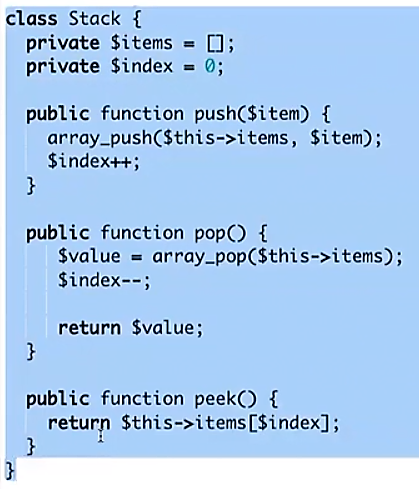
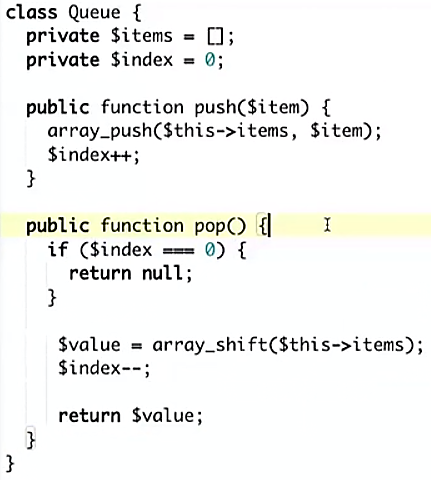
ডাটা স্টোর করার মেকানিজম হলো Data Structure

একটা প্রব্লেম সলভ করার ওয়ে হলো Algorithm

Array তে key ডুপ্লিকেট হয় না কখনো

**Abstraction প্রসেস Abstract মেথড** - যখন কোন ফাংশনের আউটপুট জানা কিন্তু ভিতর প্রসেস অজানা

**Abstract/Basic Data Structure** – Array/list (item list) , Map, Stack(LIFO), Queue(FIFO), Set (unique item list)



Excercism.org

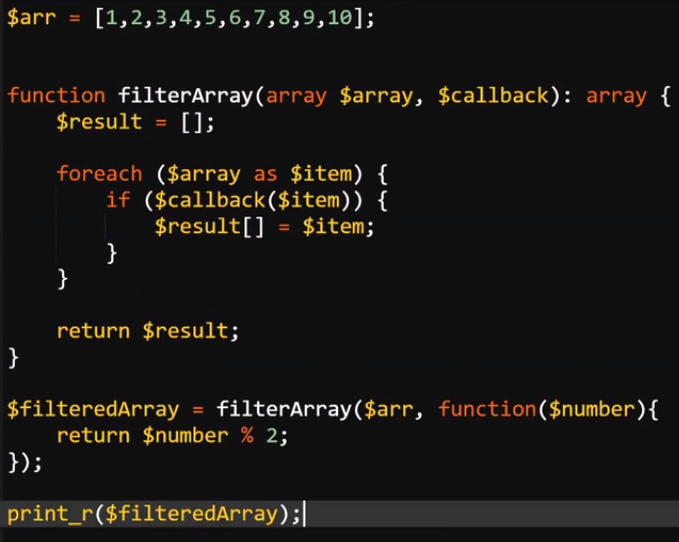
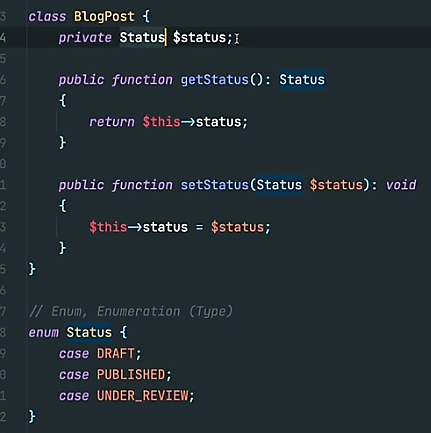
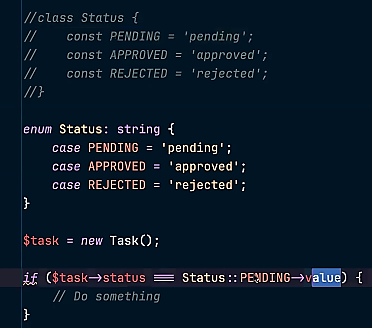
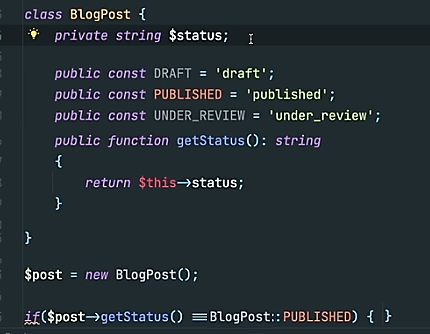
<https://www.php.net/manual/en/function.sort.php>

<https://www.php.net/manual/en/function.array-search.php>

Check sorting Algorithm (Bubble sort, selection,merge, quick sort etc)

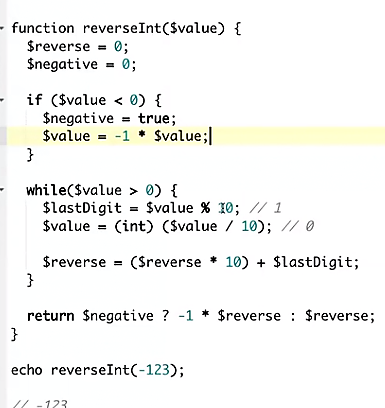
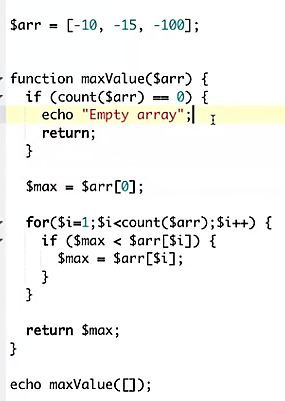
Searching Algorithm (Linear, binary, ternary )

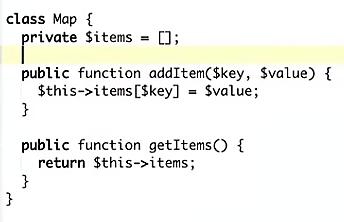
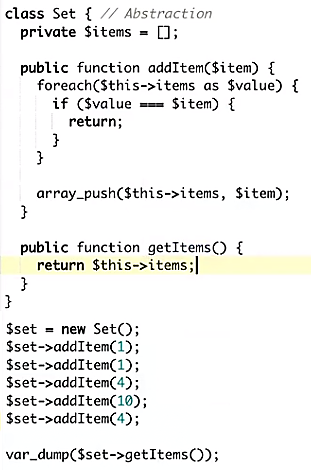
**Enum/Enumeration** sets a custom type like integer, string in PHPv8.1, has benefit to validate the options



Callback function: anonymous fx or any fx

Onlinephp.io





**R-01 : PHP Basics**

echo get\_type($var) var\_dump($var);

“” is an empty string but null is null , null != “”

**Type juggling** – Type of variable determined by the value, dynamically auto type casting when needed

“4” + 2 🡺 6

*declare(strict\_types = 1);*

**Type Hinting**: Saying variable type function age\_cal(int age)

**Type Casting**: Changing variable type

floatval(), intval(), boolval()

**Magic Method**: \_toString() 🡨 Object কে String এ রুপান্তর করা

Object কে এরে তে type casting করলে property হবে key এবং value হবে null বা দেয় value

**Variable Scope:** local, global, static ex: global $username;

**Variable Interpolation**: String এর মদ্ধে variable value কে show করা

echo “hello {name}” অথবা, echo “hello $name”;

**Array** : indexed Array, Associative Array

**Array Manipulation**: count(), sort(), array\_search($needle,$haystack,$strict), array\_reverse(), array\_unique(),

array\_unshift() 🡨 প্রথমে যোগ হবে, array\_pop() 🡨 শেষে বিয়োগ array\_shift() 🡨প্রথমে বাদ হবে

join(“-”, $array) 🡨 hello-brother

*number\_format($num, $point\_decimal, $decimal\_replace,”,”)*

time() 🡨latest Unix timestamp

file\_get\_contents($file\_path) unlink($file\_path) file\_exists($file\_path)

str\_repeat($string, 10) str\_word\_count($string)

**Encapsulation**: প্রোসেসিং হাইড করে ইমপ্লিমেন্টেশন কে দেখানো

**Named Arguments/Positional Arguments**: sending parameter value with identification in PHP 8

*gradMark($english,$bangla){…} gradMark(bangla:10, english:20); //calling*

**Closure/Anonymous function/Lambda function:** one-time function. Call-back ফাংশন হিসেবেই বেশী ব্যবহ্রিত হয়

|  |  |  |
| --- | --- | --- |
| *$yrs = [2002,2003,2004];*  *function incYr(yr){*  *Return year+10;*  *}*  *$new=array\_map( ‘incYr’, $yrs);* | *$yrs = [2002,2003,2004];*  *$nam = “Rubel”;*  *$fn = Function (yr) use ($nam){*  *Return year+10;*  *}*  *$new=array\_map( $fn, $yrs);*  *$fn(2007);* | *//Arrow function*  *Array\_map(*  *fn ($yr)=>{$yr+10},$yrs)* |

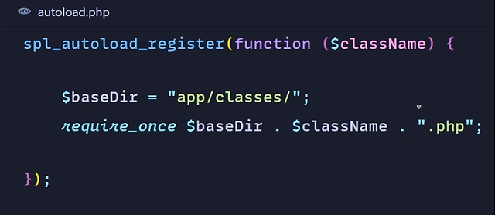
Arrow function এ যা থাকবে তাই রিটার্ন করবে, আলাদা করে return করা লাগে না

যতক্ষন পর্যন্ত condition true ততক্ষন পর্যন্ত লুপ চলবে

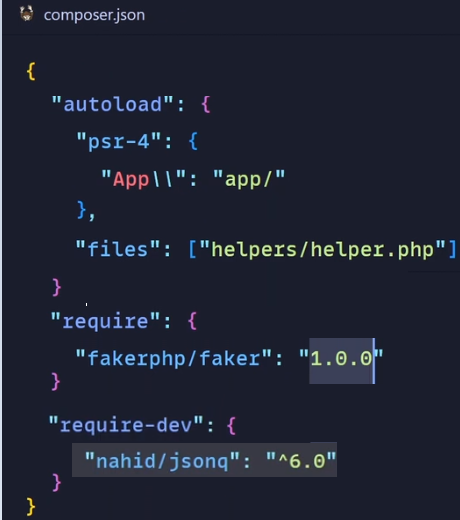
**R-02 : Autoloading and Dependency Management**

**Autoload:**

*require\_once \_\_DIR\_\_ . “/vendor/autoload.php”*



Initiate all classes from base dir/folder  
**Composer:**

*composer*

*composer init*

*composer -v*

*composer dump-autoload*

*composer show*

*composer install*

*composer install --no-dev*

*composer update*

*composer require package\_name*

*composer require package\_name --dev*

*composer require package\_name:version*

*composer remove package\_name*

*composer audit*

*composer search package\_name*

**PSR** – PHP Standards Recommendations (<https://php-fig.org/>)

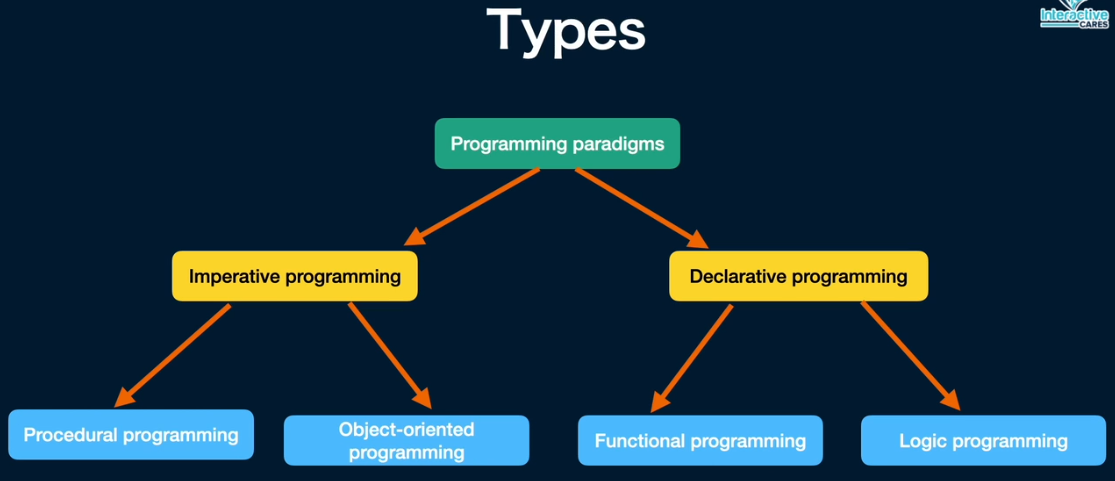
<https://devsonket.com/composer>

<https://packagist.org>

*echo PHP\_EOL;*

**R-02 : OOP in PHP**

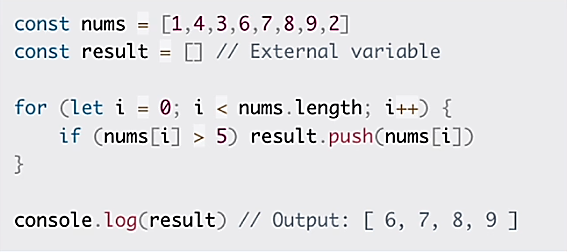
**Paradigm**: the way we see something, point of view, The way we understand or interpret something



**Imperative Programming**: Imperative comes from *impero* (“I command”). Sets of detailed instructions that are given to the computer to execute in a given order. কি এবং কিভাবে হবে সব বলে দেয়া

**Declarative Programming**: কি হবে শুধু তা বলে দেয়া । what you want your program to achieve rather than how it should run.

Imperative Programming Declarative Programming



Sql one paradigm language, JavaScript multi paradigm language

**Meta Programming/ Generic programming paradigm** : রানটাইমে নিজেকে ইভ্যালুয়েট করা কিংবা অন্য প্রোগ্রামকে রিড করতে পারে, Code as a Data

**Visibility**: Public, Private (only in class), Protected (used in inheritence)

Class এর মধ্যে Best practice হলো সব প্রোপার্টি private রাখা যাতে বাইরে থেকে চেঞ্জ না করা যায়

*class Task{*

*private static name;*

*public static function greet(){*

*}*

*}*

*Task::greet();*

static প্রোপার্টি প্রত্যেক object এ global value এর মত অপরিবর্তিত থাকে কারন তা class এ scope এর মত fix হয়

*public function \_\_destruct(){*

*// অবজেক্ট ধ্বংস হবার সময় যদি কোন কাজ করার থাকে তখন এই স্পেশাল মেথড দরকার পরে*

*}*

**Inheritance** : একটা ক্লাস/Object এর সাথে আরেকটা ক্লাস/Object এর “is a” রিলেশনশিপ।

Car is a Vehicle 🡺 class Car extends Vehicle{ }

**Abstruct Class & Abstruct Method:**

abstract Method (No implementation/body) is in abstract class

abstruct class Vehicle{

abstruct public function getBaseFare():int;

public function getTotal(){ //same formula for all child }

}

Class Car extends Vehicle{

Public function getBaseFare(){ return 50; }

}

Class Bike extends Vehicle{

Public function getBaseFare(){ return 20; }

}

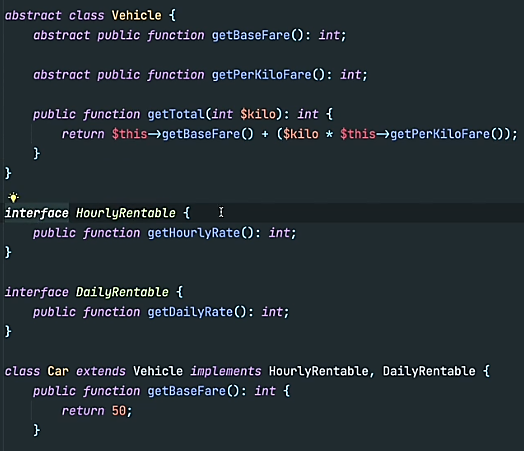
Abstruct method যুক্ত ক্লাসকে যদি extends করা হয় তাহলে প্রত্যেক child ক্লাসকে অবশ্যই abstruct method বডি ইমপ্লিমেন্ট করা লাগবে ।

child class এর মেথডের রিটার্ন ভিন্নতার কারনেই parent class এ abstruct method এর প্রয়োজন হয় আর abstruct method থাকার জন্যই parent class abstruct class এ পরিনত হয় ।

Abstruct class = Abstruct Method + Concrete method

Interface = Method doesn’t have concrete implemenatation & body

একটা Interface এ সব মেথডই abstruct method (No Body), তাই আলাদা করে abstruct কীওয়ার্ডের প্রয়োজন নেই



এমন কোন ফিচার যা সব child class এর দরকার নেই তখন সেগুলো আলাদা আলাদা interface হিসেবে প্রয়োজনমাফিক child class এ implements করা যেতে পারে

**Polymorphism:**

Polymorphism runs on 2 terms:

1. **Compile time polymorphism – method Overloading** (PHP not support it)

Decide same Method in compile time based on unequal parameters



1. **Run time polymorphism – method Overriding**

Method rewrite in child class with different functionality. Decision made in runtime.

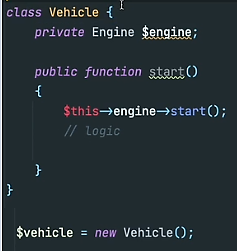
**R-03 : PHP Advanced Topics**

**Namespaces**: established in PHPv5.3. prevent naming duplicity.

*Namespace App\route.php*



**Composition**: একটা object/class এর মধ্যে এক/একাধিক object/class কে ব্যবহার করা। code reusable.

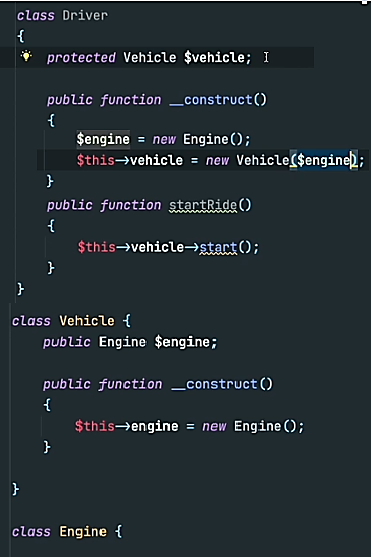
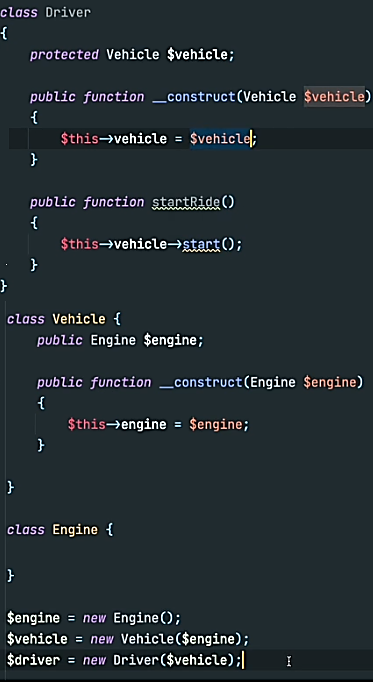


Composition “has a” relationship - ex: Vehicle has a Engine

Inheritance “is a ” relationship -ex: Car is a Vehicle

**Dependency Injection**: উপরের class based উদাহরনে ,Vehicle class টি Engine class এর উপর ডিপেন্ডেন্ট।

multiple dependency can be in multiple layer in multiple class

**Tightly coupled Dependency** – instantiation process is limited only on current class এবং চেঞ্জ করার সুযোগ কম। যদি এই dependency class এর আরো কোন dependency থাকে তাও দিয়ে দিতে হচ্ছে।

**Loosely coupled Dependency** – ক্লাসে dependency কিভাবে তৈরি হবে এবং তার আরো কোন nested dependency আছে কিনা তা জানা লাগলো না. Dependency বাহির থেকে প্রোভাইড করা

Setter injection – dependency injected in setter method

constructor injection - dependency injected in constructor method

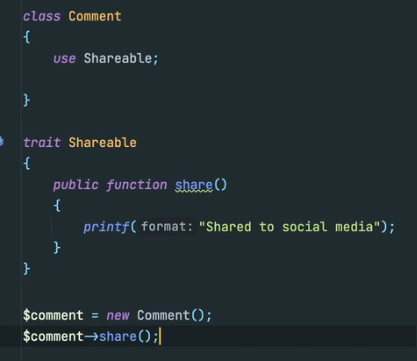
**OOP design principles**

**Program to an interface**: tightly coupled/hard coded class dependency কে তার বিহেভিয়ার অনুযায়ী child class সাপোর্টের জন্য interface/abstruct class এর উপর ডিপেন্ডেন্ট করা

**Encapsulate what varies** : প্রোগ্রামের যে পার্ট প্রতিনিয়ত পরিবর্তনশীল তাকে অপরিবর্তনীয় কোড থেকে আলাদা করা । class label অথবা method label ভেঙে ভেঙে isolate করা।

**Traits**: functionality mixup in a class/object. কোন Behaviour কে ক্লাসগুলো নিজেদের মধ্যে শেয়ারের পদ্ধতি

যে ফাংশনগুলোর Has a (Composition) / is a (Inheritence) কোন রিলেশন নেই বা independent



**Magic method**:

\_\_construct(){} \_\_destruct(){} \_\_get(){} \_\_set(){} \_\_isset(){} \_\_unset(){}

\_\_toString(){} \_\_serialize(){} \_\_unserialize(){} \_\_invoke(){} \_\_debugInfo(){}

\_\_call(){ $method, $args } 🡪 object এর কোন মেথড না পাওয়া গেলে তখন এটা অটো কল হয়।

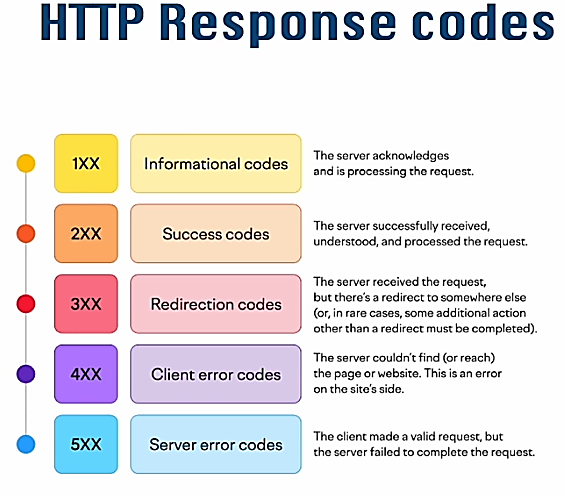
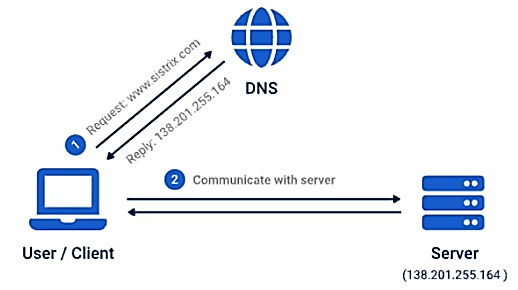
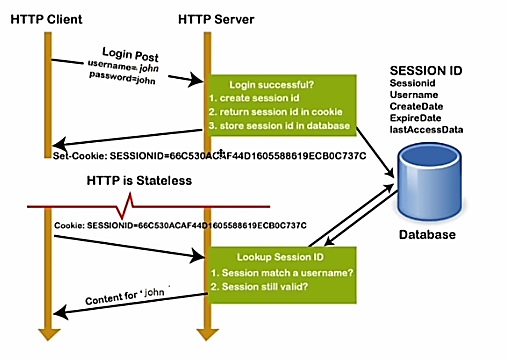
\_\_callStatic(){ $method, $arg } 🡪 কোন static মেথড না পাওয়া গেলে তখন এটা অটো কল হয়।

For more info: <https://www.php.net/manual/en/language.oop5.magic.php>

**R-05 : Web Primer**

Internet – Interconnected Network , infrastructure

Web – service from internet DNS – Domain Name Service



**R-06 : Intro to Framework**

Micro framework: slim, flask, bottle, express js

Choosing which is better Framework or CMS ?

1. Requirements
2. Cost
3. Deadline
4. Scalability: traffic, update

**R-07 : How to build an MVC Framework**

$connection = mysqli\_connect(db\_host, db\_user, db\_pass);

mysqli\_select\_db($db\_name, $connection);

$query = “select \* from settings”;

$result = mysqli\_query($connection, $query);

while($row = mysqli\_fetch\_assoc($result)){ ……. }

php -S localhost:80

php -S localhost:80 -t public

Steps:

1. Set metadata like description, name etc in composer.json (may use composer init)
2. Create folders & files with corresponding code
3. Public 🡪 css, js, img, index.php
4. App 🡪 controllers (base controller.php), models(base model.php), routes (web.php)
5. Set autoload of classes folder, namespace and helper file in composer.json
6. connect vendor autoload in index.php
7. add .htaccess in public folder

<IfModule mod\_rewrite.c>

RewriteEngine On

RewriteRule ^ index.php [QSA,L]

</IfModule>

1. update routes/web.php [here App\Controller is namespace]

return [

"/profile" => ["App\Controllers\ProfileController","index"],

"/profile/edit" => [ProfileController::class,"edit"]

];

1. update index.php for controlling routes

$routes = require\_once \_\_DIR\_\_."/../routes/web.php";

$request = explode('?',$\_SERVER['REQUEST\_URI'])[0];

if(isset($routes["$request"])){

$controller = $routes["$request"][0] ?? null;

$action = $routes["$request"][1] ?? null;

if($controller && $action){

$newController = new $controller();

$newController->$action();

}

}else{

throw new exception("404 Not Found");

}

1. helper for view() function

function view($view,$data=[]){

extract($data);

require \_\_DIR\_\_."/../resources/views/{$view}.php";

}

1. save db configuration info in config/database.php file
2. establish PDO in base Model

protected PDO $db;

public function \_\_construct(){

$config = require\_once \_\_DIR\_\_."/../../config/database.php";

try{

$this->db = new PDO("mysql:host={$config['host']};dbname={$config['dbname']}",$config['username'],$config['password']);

$this->db->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_EXCEPTION);

$this->db->setAttribute(PDO::ATTR\_DEFAULT\_FETCH\_MODE,PDO::FETCH\_ASSOC);

}catch(\PDOException $e){

echo $e->getMessage();

}

}

1. Fetch DB data from controller and pass it to view to show

**LR-02 : How to build a simple PHP CLI application**

PHP রান করা যায় দুই ভাবে।

* 1. ওয়েব সার্ভার থেকে । web server থেকে রান করালে রানিং web server এর সাথে php কনফিগার করা লাগে, তখন request web server এর মাধ্যমে php হ্যান্ডেল করে।
  2. CLI থেকে।

CLI তে http request বলে কিছু নেই

CLI tools – rector (upgrade & automate), phpstan(Bug finder), php\_codesniffer,

*php file\_name.php* //run a php file in console

*php file\_name.php min=1 max=20* //sending arguments

*chmod +x file\_name.php* //change permission to executable

*where php*

count = $argc

filename=$argv[0] min=$argv[1] max=$argv[2]

***#! /usr/bin/php*** //shebang -interpreter location

***#! /usr/bin/env php*** //tell environment to find php

***//get input from user***

***$pass = readline(“password: ”);*** //return string

**$options = getopt(‘’,[‘min::’,’max:’])** //return array, max is must but min is optional

**$pass = fscanf(STDIN, ‘%d’, $input);**

C, Rust, Go বেইজড CLI অ্যাপ্লিকেশন ভালো পারফরমেন্স দেয়।

**Package for build CLI tool using php**: symphony console components, minicli-php, Laravel zero