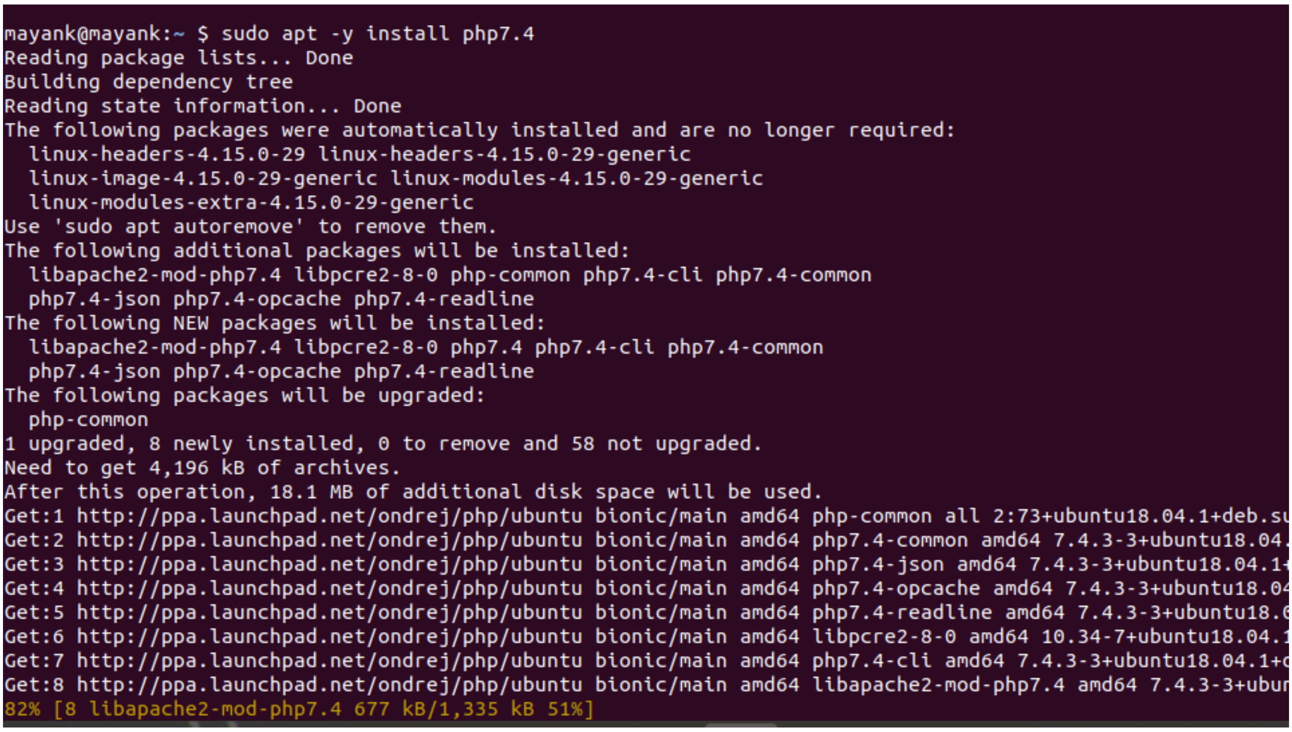
PHP DESIGN PATTERN ASSIGNMENT

Q1:- update your php to version 7.4

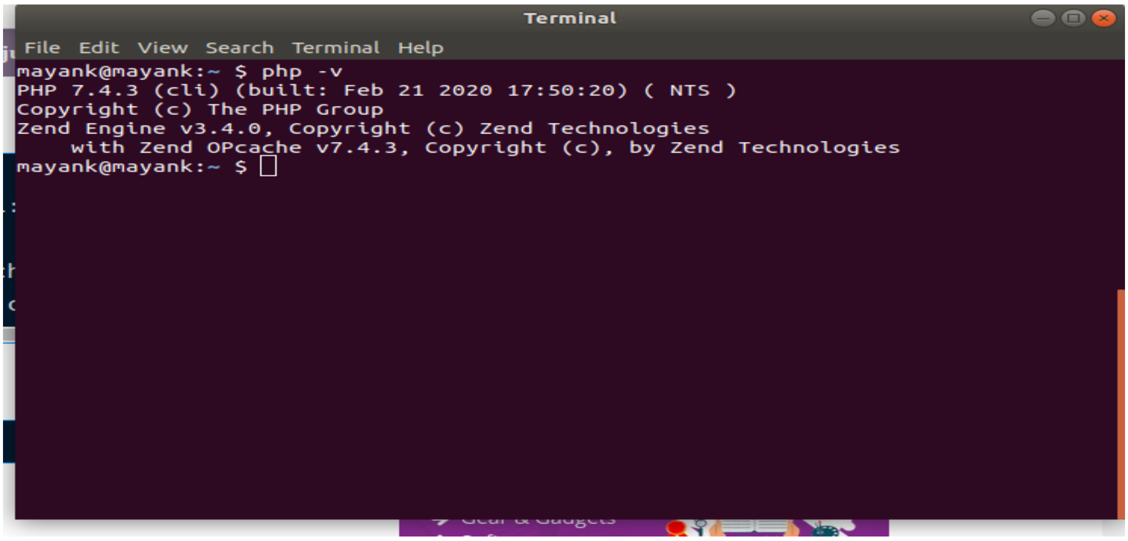
soln:-

Step 1:- sudo apt -y install php 7.4



```
mayank@mayank:~ $ sudo apt -y install php7.4
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-29 linux-headers-4.15.0-29-generic
  linux-image-4.15.0-29-generic linux-modules-4.15.0-29-generic
  linux-modules-extra-4.15.0-29-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libapache2-mod-php7.4 libpcre2-8-0 php-common php7.4-cli php7.4-common
  php7.4-json php7.4-opcache php7.4-readline
The following NEW packages will be installed:
  libapache2-mod-php7.4 libpcre2-8-0 php7.4 php7.4-cli php7.4-common
  php7.4-json php7.4-opcache php7.4-readline
The following packages will be upgraded:
  php-common
1 upgraded, 8 newly installed, 0 to remove and 58 not upgraded.
Need to get 4,196 kB of archives.
After this operation, 18.1 MB of additional disk space will be used.
Get:1 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 php-common all 2:73+ubuntu18.04.1+deb.su
Get:2 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 php7.4-common amd64 7.4.3-3+ubuntu18.04.
Get:3 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 php7.4-json amd64 7.4.3-3+ubuntu18.04.1+
Get:4 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 php7.4-opcache amd64 7.4.3-3+ubuntu18.04
Get:5 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 php7.4-readline amd64 7.4.3-3+ubuntu18.0
Get:6 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 libpcre2-8-0 amd64 10.34-7+ubuntu18.04.1
Get:7 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 php7.4-cli amd64 7.4.3-3+ubuntu18.04.1+c
Get:8 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 libapache2-mod-php7.4 amd64 7.4.3-3+ubun
82% [8 libapache2-mod-php7.4 677 kB/1,335 kB 51%]
```

step2 : check php version php -v

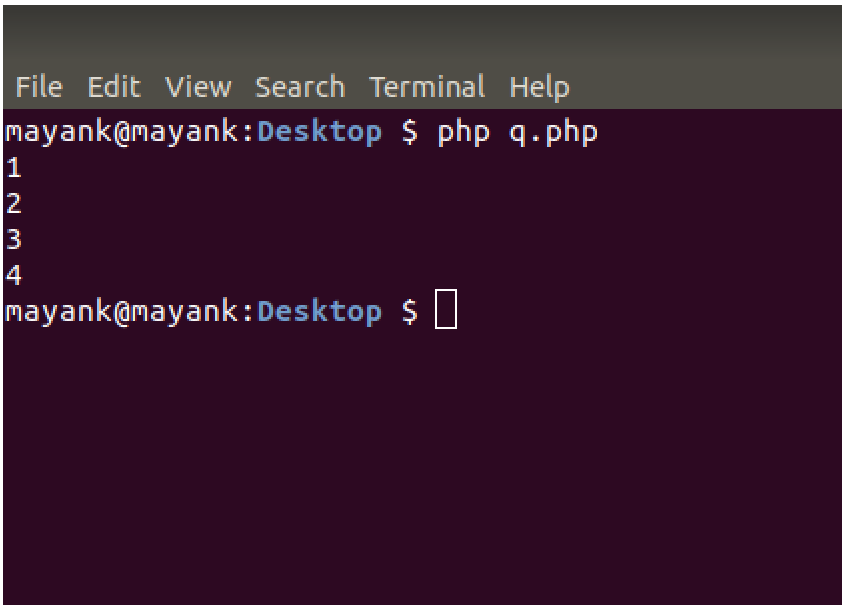


```
mayank@mayank:~ $ php -v
PHP 7.4.3 (cli) (built: Feb 21 2020 17:50:20) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
mayank@mayank:~ $
```

Que 2:- what are the advantages of spread operator over array_merge in php ?

Soln :- Array spread operator does the same thing as array merge . Using array spread operator we can directly merge an array elements into another array elements .
Example of array spread is as follow:-

```
<?php
ini_set('display_errors',1);
$a = [1, 2];
$b =[...$a ,3,4];
$len=count($b);
for($i=0;$i<$len;$i++)
{
echo"$b[$i]";
echo"\n";
}
?>
```



```
mayank@mayank:Desktop $ php q.php
1
2
3
4
mayank@mayank:Desktop $ 
```

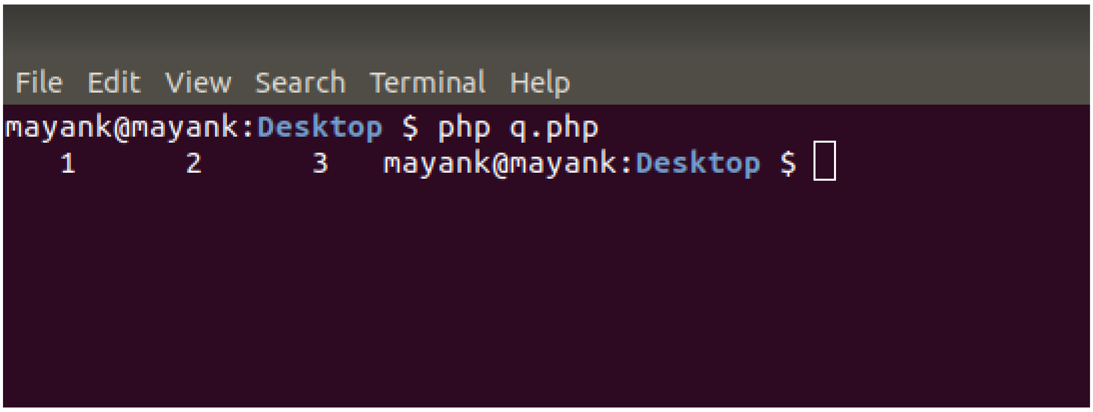
the advantage of array spread over array merge is that it save the time for the merging condition the time and space complexity of the code is reduced . And we get the desired result in fast and efficient manner.

Que 3 :- Write the output of these;

```
$arr1 = [1, 2, 3];

$arr2 = [...$arr1];

$arr3 = [0, ...$arr1];

$arr4 = array(...$arr1, ...$arr2, 111);
```

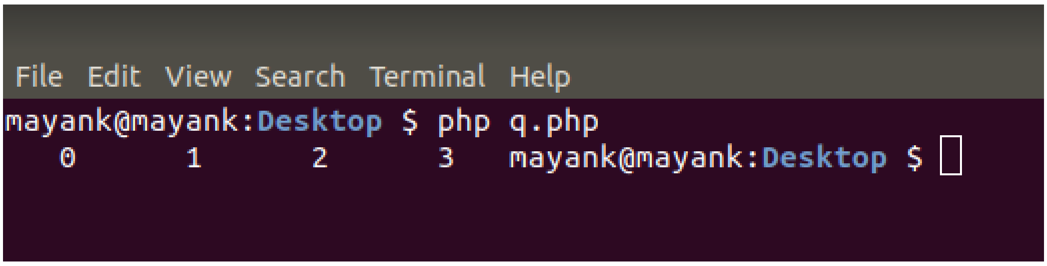
what will be the output of array $arr2 , $arr3 and $arr4 soln:-

output of arr2 :- 123 as $arr1 is merged in a blank arrray



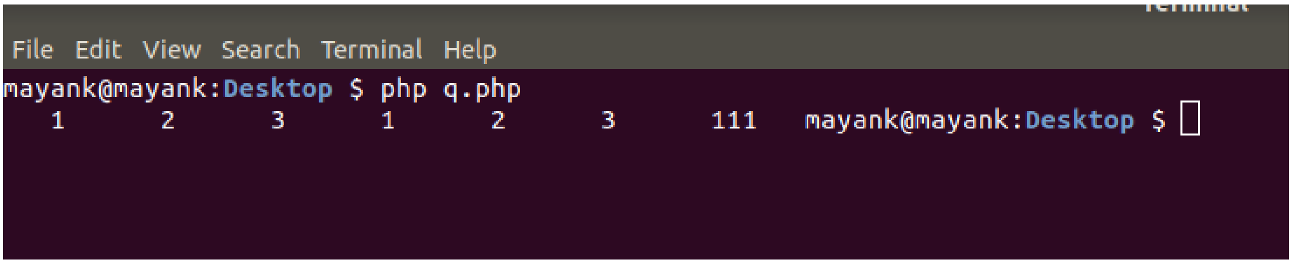
```
mayank@mayank:Desktop $ php q.php
   1      2      3   mayank@mayank:Desktop $ 
```

output of array arr3 :- 0 1 2 3 as $arr1 merged after 0



```
mayank@mayank:Desktop $ php q.php
   0      1      2      3   mayank@mayank:Desktop $ 
```

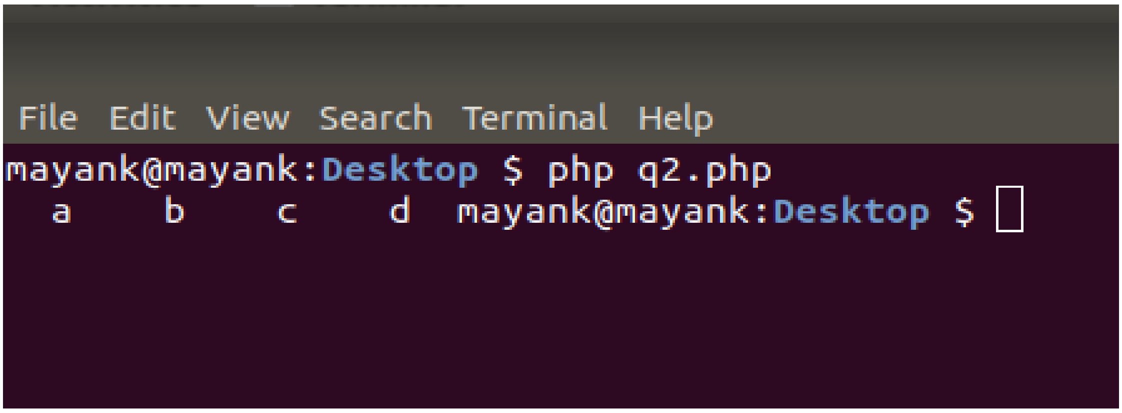
Output of arrray 4:- 1 2 3 1 2 3 111 as $ar1 and $ arr2 have same values and placed before 111



```
mayank@mayank:Desktop $ php q.php
   1      2      3      1      2      3      111   mayank@mayank:Desktop $ 
```

Ques 4 :- Write a program to print an array in which output is returned by function using the spread operator.

```
Soln:-  <?php
$arr1=['c','d'];
function showarr()
{
return ['a','b'];
}
$arr2=[...showarr(),...$arr1];
$len=count($arr2);
for($i=0;$i<$len;$i++)
{
echo" $arr2[$i] ";
}
?>
```



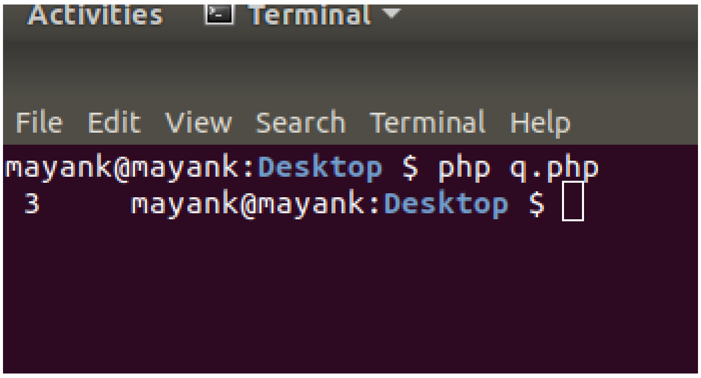
```
mayank@mayank:Desktop $ php q2.php
  a    b    c    d  mayank@mayank:Desktop $ 
```

Ques 5:- What is dependency Injection?

Soln:- Dependency Injection:-
Dependency Injection in simple way is the design pattern that helps avoiding hard copied dependencies for some piece of codes or hardwares .

1. Constructor dependency :- allows to inject the dependency through constructor and object of a class .

```
<?php
class Programmer {
private $skills;
public function __construct($skills){
$this->skills = $skills;
}
public function totalSkills(){
return count($this->skills);
}
}
$createskills = array("PHP", "JQUERY", "AJAX");
$p = new Programmer($createskills);
echo " " ;
echo $p->totalSkills()." ";
?>
```

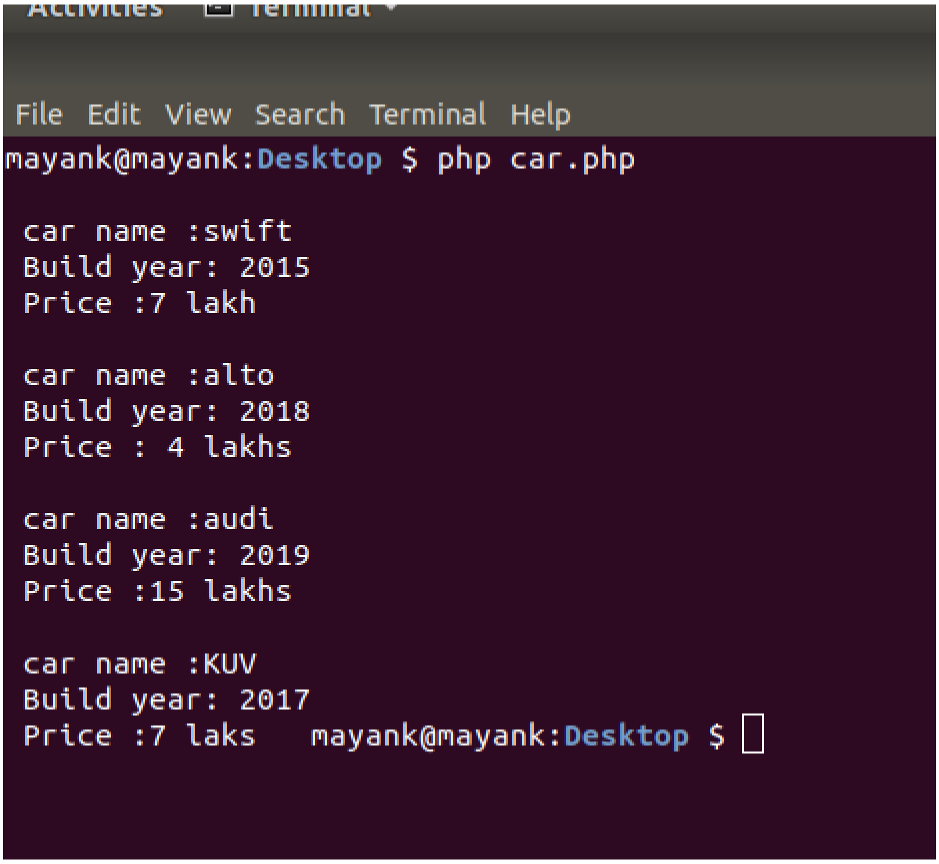


```
mayank@mayank:Desktop $ php q.php
 3     mayank@mayank:Desktop $ 
```

Ques 6. Write an example of a factory class where we pass 4 different car models and it returns price and builds year of the car.

```
Soln <?php
class Cars
{
private $carname;
private $year;
private $price;
public function __construct($name)
{
$this->carname=$name;
if($this->carname=="audi")
{
$this->year= 2019;
$this->price="15 lakhs ";
}
elseif($this->carname=="swift")
{
$this->year=2015;
$this->price="7 lakh ";
}
elseif($this->carname=="KUV")
{
$this->year=2017;
$this->price="7 laks ";
}
elseif($this->carname=="alto")
{
$this->year=2018;
$this->price=" 4 lakhs ";
}
}
public function display()
{
echo "\n car name :".$this->carname;
echo "\n Build year: ".$this->year;
echo "\n Price :".$this->price;
}
}
class carFactory
{
public static function create($name)
{
return new Cars($name);
}
}
$test=carFactory::create("swift");
$test1=carFactory::create("alto");
$test2=carFactory::create("audi");
$test3=carFactory::create("KUV");
```

```
print($test->display());
echo"\n";
print($test1->display());
echo"\n";
print($test2->display());
echo"\n";
print($test3->display());
?>
```



```
mayank@mayank:Desktop $ php car.php

 car name :swift
 Build year: 2015
 Price :7 lakh

 car name :alto
 Build year: 2018
 Price : 4 lakhs

 car name :audi
 Build year: 2019
 Price :15 lakhs

 car name :KUV
 Build year: 2017
 Price :7 laks   mayank@mayank:Desktop $ 
```

Ques 7:-Give an example of singleton class.

Soln: A singleton class is a class that can be called by only a single object or we can say that the classes that can have only a single intsance are called singleton classes

an example of singleton classes is given below:-

 a singleton class to database connectivity:-

```
<?php
class ConnectDbWOSingleton {
private $conn;
private $host = 'localhost';
private $user = 'db user-name';
private $pass = 'db password';
private $name = 'db name';
// Public constructor.
public function __construct()
{
$this->conn = new PDO("mysql:host={$this->host};
dbname={$this->name}", $this->user,$this->pass,
array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES 'utf8'"));
}
public function getConnection()
{
return $this->conn;
}
}

$instance = new ConnectDbWOSingleton();
$conn = $instance->getConnection();
var_dump($conn);
$instance = new ConnectDbWOSingleton();
$conn = $instance->getConnection();
var_dump($conn);
$instance = new ConnectDbWOSingleton();
$conn = $instance->getConnection();
var_dump($conn);
?>
```

Ques 8:- What are the benefits of following design patterns?
Soln:

Design patterns are commonly defined as time-tested solutions to recurring design problems. The term refers to both the description of a solution that you can read, and an instance of that solution as used to solve a particular problemDesign patterns have two major benefits:-

1. they provide you with a way to solve issues related to software development using a proven solution. The solution facilitates the development of highly cohesive modules with minimal coupling. They isolate the variability that may exist in the system requirements, making the overall system easier to understand and maintain.

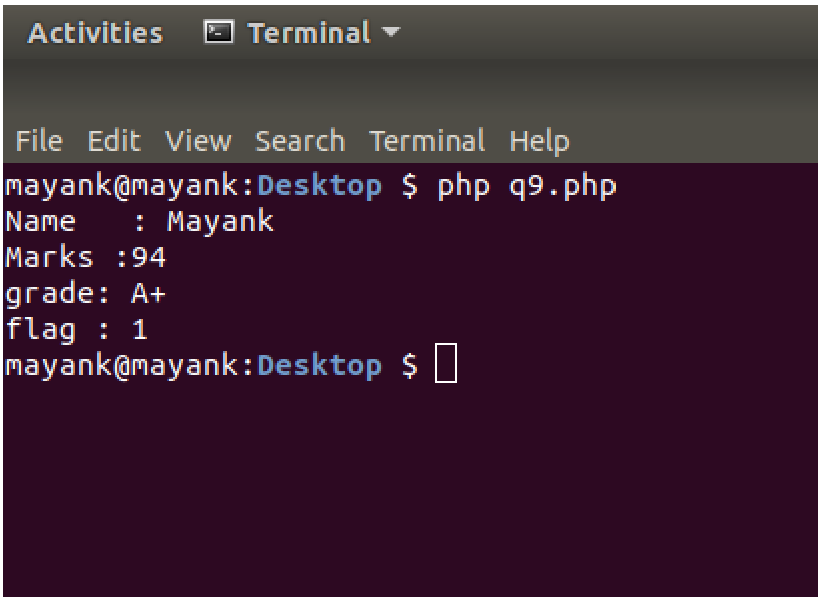
2.Design patterns make communication between designers more efficient. Software professionals can immediately picture the high-level design in their heads when they refer the name of the pattern used to solve a particular issue when discussing system design.

Ques 9:- Define a class with type properties.

Soln :-

```
<?php
class Student
{
public String $name;
var int $marks;
public String $grade;
public bool $flag;
public function __Construct(String $name ,int $marks, String $grade ,bool $flag)
{
$this->name=$name;
$this->marks=$marks;
$this->grade=$grade;
$this->flag=$flag;
}
}
$obj=new student("Mayank",94,"A+",true);
```

```
echo"Name : "."$obj->name \n";
echo "Marks :".$obj->marks."\n";
echo "grade: "."$obj->grade \n";
echo "flag : "."$obj->flag \n";
?>
```



```
mayank@mayank:Desktop $ php q9.php
Name    : Mayank
Marks :94
grade: A+
flag : 1
mayank@mayank:Desktop $
```

Ques :10 Write a function using arrow function array_map.

Soln :-

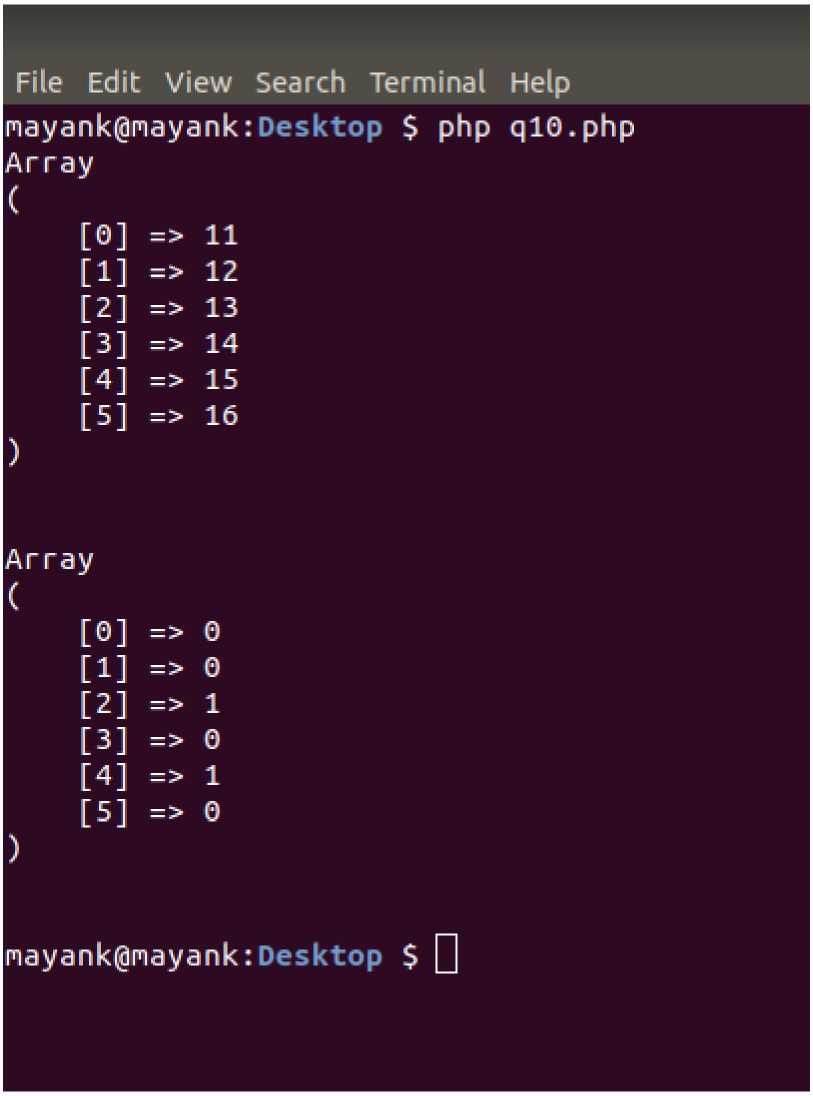
The array_map() is an inbuilt function in PHP and it helps to modify all elements one or more arrays according to some user-defined condition in an easy manner. It basically, sends each of the elements of an array to a user-defined function and returns an array with new values as modified by that function.

Syntax of array_map() function is as follow:-
array_map(functionname (), array 1 , array2.......)

example of array map is as follow:-

```
<?php
function f1($v)
{
return ($v+10);
}
function f2($v1,$v2)
{
if($v1==$v2)
return 1;
else return 0;
}
$arr1=array(1,2,3,4,5,6);
$arr2=array(2,3,3,1,5,7);
print_r(array_map("f1",$arr1));
```

```
echo"\n \n";
print_r(array_map("f2",$arr1,$arr2));
echo"\n \n";
?>
```



```
mayank@mayank:Desktop $ php q10.php
Array
(
    [0] => 11
    [1] => 12
    [2] => 13
    [3] => 14
    [4] => 15
    [5] => 16
)


Array
(
    [0] => 0
    [1] => 0
    [2] => 1
    [3] => 0
    [4] => 1
    [5] => 0
)


mayank@mayank:Desktop $ 
```