

## PHP Question 1.1

Consider that SupportBase is declared elsewhere and successfully loaded by the autoloader. Given the described class definitions, what is the output of index.php?

Application/Module/Widget.php

```
1 <?php
2
3 class Widget
4 {
5     public function showInfo($separator = null) {
6         $separator = !empty($separator) ? $separator : ':';
7         return __FUNCTION__ . $separator. __LINE__;
8     }
9 }
10
```

Support/Module/Widget.php

```
1 <?php
2
3 class Widget extends SupportBase
4 {
5     public function showInfo($separator = '-') {
6         $separator = isset($separator) ? $separator : '|';
7         return __CLASS__ . $separator . __LINE__;
8     }
9 }
10
```

index.php

```
1 <?php
2
3 require_once 'Application/Module/Widget.php';
4 require_once 'Support/Module/Widget.php';
5
6 $appWidget = new Widget();
7 $supportWidget = new Widget();
8 echo $appWidget->showInfo(' ');
9 echo $appWidget->showInfo(null);
10
```

### PHP Answer 1.1

(Provide your answer below. Use as much space as necessary.)

The given code doesn't work the right way because the Widget class names are declared the same way, there will be a conflict in names. Also I would rather use namespaces instead of require\_once.

Moreover this kind of naming Widget class will cause future misunderstandings between teammates. So I decided to do the following:

```
class Widget
{
    public function showInfo($separator = null) {
        $separator = !empty($separator) ? $separator : ':';
        return __FUNCTION__ . $separator . __LINE__;
    }
}

require_once './SupportBase.php';
class SupportWidget extends SupportBase
{
    public function showInfo($separator = '-') {
        $separator = isset($separator) ? $separator : '|';
        return __FUNCTION__ . $separator . __LINE__;
    }
}

class SupportBase{
    function test(){
        return;
    }
}

require_once './Application/Module/Widget.php';
require_once './Support/Module/SupportWidget.php';

$appWidget = new Widget();
$supportWidget = new SupportWidget();

echo $appWidget->showInfo(' ');
echo $appWidget->showInfo(null);
```

The output would be like:

showInfo 7

showInfo:7

where the 7 comes from number of the code line it was called from. 'showInfo' is name of the function.

### PHP Question 1.2

Consider that the Exam class will be successfully loaded by the autoloader.

Given the described class definition, what is the output of index.php?

## Exam.php

```
1  <?php
2
3  class Exam
4  {
5      static $defaultText = 'unknown';
6      protected $score;
7      public $place = 'office';
8      private $attributes = array();
9
10
11     function setAttributes(array $attributes) {
12         $this->attributes = (array)$attributes;
13     }
14
15     function __get($name) {
16         if (array_key_exists($name, $this->attributes)) {
17             return $this->attributes[$name];
18         }
19         else {
20             return static::$defaultText;
21         }
22     }
23 }
```

index.php

---

### PHP Answer 1.2

(Provide your answer below. Use as much space as necessary.)

First of all the syntax error: `$exam->score = ;`

The second thing is the `$score` property cannot be accessed in `index.php` because it's protected. The way we could access it is using a get method like `setAttributes()`

So I would do the following changes:

```
require_once 'Exam.php';
```

```
$attributes = ['time' => date(time()), 'place' => 'Sonny\'s', 'timeLimit' => null];
```

```
$exam = new Exam();
```

```
$exam->setAttributes($attributes);
```

```
$exam->score = 5; // should be accessed via a getter method declared in Exam.php
```

```
echo $exam->time . "\n";
```

```
echo $exam->place . "\n";
```

```
echo $exam->timeLimit . "\n";
```

```
echo $exam->candidate . "\n";
```

### PHP Question 1.3

Assume that the Greeter and InformalGreeter classes will be successfully loaded by the autoloader. Given the described class definitions, what is the output of index.php?

Greeter.php

```

1  <?php
2
3  abstract class Greeter
4  {
5      private $target = 'World';
6
7      public function getTarget() {
8          return $this->target;
9      }
10
11     public function sayHello(string $punctuation = '!') {
12         echo 'Hello, ' . $this->target . $punctuation;
13     }
14 }

```

InformalGreeter.php

```

1  <?php
2
3  class InformalGreeter extends greeter
4  {
5      public function sayHello() {
6          echo 'Hi, ' . $this->target . '!';
7      }
8  }

```

index.php

```

1  <?php
2
3  $greeter = new Greeter();
4  $informalGreeter = new InformalGreeter();
5
6  echo "Greeting $greeter->getTarget() ...";
7  $greeter->sayHello('?');
8  echo "\n\n\n";
9
10 echo "Greeting $informalGreeter->getTarget() ...";
11 $informalGreeter->sayHello();
12 echo "\n\n\n";

```

### PHP Answer 1.3

(Provide your answer below. Use as much space as necessary.)

We can't create an instance of abstract class like `new Greeter()`. Also we should follow the abstract class parameters in order to override the `InformalGreeter` class methods. The next we have to do is change the `$target` property to protected so that it will be accessible in `InformalGreeter` class.

The way the given code would work after some changes is:

```
<?php
```

```

abstract class Greeter
{
    protected $target = 'World';
    public function getTarget() {
        return $this->target;
    }

    public function sayHello(string $punctuation = '!') {
        echo 'Hello, ' . $this->target . $punctuation;
    }
}

class InformalGreeter extends Greeter
{
    public function sayHello(string $punctuation = '!') {
        echo 'Hi, ' . $this->target . '!';
    }
}

require_once 'Greeter.php';
require_once 'InformalGreeter.php';

// $greeter = new Greeter();
$informalGreeter = new InformalGreeter();

// echo "Greeting $greeter->getTarget() ...";
// $greeter->sayHello('?');
// echo "\n\n\n";

echo "Greeting " . $informalGreeter->getTarget() . " ...";
$informalGreeter->sayHello();
echo "\n\n\n";

```

The output will be like: Greeting World ...Hi, World!

### PHP Question 2.3

Without altering the given code:

- Represent a universal remote that can turn on a television and a cable box.
- Represent a remote-control user who can push the `power on` button of a remote control.
- Have a remote-control user push the `power on` button of a tv remote.
- Have the same remote-control user push the `power on` button of a universal remote.

```
<?php

interface RemoteControl {
    public function powerOn();
}

class Television {
    public function setPower($state) {
        // Truncated Code
    }
}

class CableBox {
    public function setPower($state) {
        // Truncated Code
    }
}

class TVRemote implements RemoteControl
{
    private $tv;

    public function __construct(Television $tv) {
        $this->tv = $tv;
    }

    public function powerOn() {
        $this->tv->setPower(true);
    }
}
```

### PHP Answer 2.3

(Provide your answer below. Use as much space as necessary.)

1.

```
class UniversalRemote extends TVRemote
{
    private $cableBox;

    public function __construct(CableBox $cableBox) {
        parent::__construct(new Television());
        parent::powerOn();
        $this->cableBox = $cableBox;
    }

    public function powerOn() {
        $this->cableBox->setPower(true);
    }
}
```

2.

```
class User extends TVRemote
{
    public function __construct() {
        $this->tv = new Television();
        parent::__construct($this->tv);
    }

    public function powerOnButton() {
        $this->powerOn();
    }
}
```



3.

```
65         $this->powerOn();
66     }
67 }
68
69 $user = new User();
70 $user->powerOnButton();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

We're in Matrix.

4.

```
class User extends TVRemote
{
    public function __construct() {
        $this->tv = new Television();
        parent::__construct($this->tv);
    }

    public function powerOnButton() {
        $this->powerOn();
    }

    public function powerOnUniversalButton(UniversalRemote $universalRemote) {
        $universalRemote->powerOn();
    }
}
```

```
76
77 $user = new User();
78 $cableBox = new CableBox();
79 $user->powerOnUniversalButton(new UniversalRemote($cableBox));|
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

We're in Matrix.

#### PHP Question 2.4

Provide a Horse, Bird, and Pegasus class given the following rules:

- A Horse gallops.
- A Bird flies.
- A Pegasus flies in *exactly* the same way as a Bird.
- A Pegasus gallops in *exactly* the same way as a Horse.

**Note:** Since galloping and flying are complicated operations, they are assumed to require several lines of non-trivial code. The details of these algorithms are irrelevant to this question. You may represent them in your code using comments. For example:

```
function fly() {  
    // flying code here.  
}
```

#### PHP Answer 2.4

(Provide your answer below. Use as much space as necessary.)

```
interface IAnimal {  
  
}  
  
interface IBird extends IAnimal {  
  
}  
  
interface IHorse extends IAnimal {  
  
}
```

```

interface IPegasus extends IBird,IHorse{
}

class Bird implements IBird {
    public function fly() {
        echo 'Look who\'s flying now! ' . __CLASS__;
    }
}

class Horse implements IHorse{
    public function gallop() {
        echo 'Look who\'s galloping now! ' . __CLASS__;
    }
}

class Pegasus implements IPegasus {
    public function fly() {
        echo 'Look who\'s flying now! ' . __CLASS__;
    }
    public function gallop() {
        echo 'Look who\'s galloping now! ' . __CLASS__;
    }
}

$pegasus = new Pegasus();
$bird = new Bird();
$horse = new Horse();

echo $pegasus->fly() . "\n";
echo $pegasus->gallop() . "\n";
echo $bird->fly() . "\n";
echo $horse->gallop() . "\n";

```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
			<pre> Look who's flying now! Pegasus Look who's galloping now! Pegasus Look who's flying now! Bird Look who's galloping now! Horse </pre>

# MySQL

## MySQL Question 1

Considering the query below, if you needed to add the **tbl3** slug field into the result set how would you do it?

```
<?php
$pdo = new PDO('user', 'pass', 'localhost');
$pdo->exec("
    SELECT tbl1.name, tbl2.id
    FROM tbl1
    JOIN tbl2 ON tbl2.fk_tbl1_id = tbl1.id
    WHERE tbl2.fk_tbl1_id NOT IN (SELECT id FROM tbl3 WHERE name NOT IN ('name1', 'name2'))
    ORDER BY tbl1.name ASC;
");
```

## MySQL Answer 1

(Provide your answer below. Use as much space as necessary.)

I think we have to add an additional JOIN like:

```
SELECT tbl1.name, tbl2.id, tbl3.slug
```

```
FROM tbl1
```

```
JOIN tbl2 ON tbl2.fk_tbl1_id = tbl1.id
```

```
JOIN tbl3 ON tbl3.fk_tbl1_id = tbl1.id
```

```
WHERE tbl3.name IN ('name1', 'name2')
```

```
ORDER BY tbl1.name ASC;
```

## MySQL Question 2

The following query throws a duplicate primary key error and blocks a cron job to move forward. How can this query be changed so the error is not thrown, and the record is updated?

```
<?php
    $pdo = new PDO('user', 'pass', 'localhost');
    $pdo->exec("
        INSERT INTO tbl1 (id, name) VALUES (231, 'Item 231 new value.');
```

## MySQL Answer 2

(Provide your answer below. Use as much space as necessary.)

We have to check if the record exists first, if it does – UPDATE, if it does not exist then INSERT.

# Linux

## Linux Question 1

The PHP application running is throwing a massive number of errors, warnings, notices on the logs and you need to try to look for a specific error on the error logs. On the command line how can you achieve this?

## Linux Answer 1

(Provide your answer below. Use as much space as necessary.)

Something like this: `grep "specific error" *.log`

## Linux Question 2

Support calls and informs that the PHP application site is running very slow and hanging. The users can't login or even logout. A page request takes so long that Apache throws a timeout error. Considering that you have a standard PHP application and that it connects to a MySQL database that lives in another server and all your apache, MySQL and PHP configurations are correct, what steps would you take to troubleshoot the issue?

## Linux Answer 2

(Provide your answer below. Use as much space as necessary.)

I would check the apache error log files like this: `sudo tail -f /var/log/apache2/error`