

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!