

## Sections

### Exercise 00: Short and proud

Exercise 00: Short and proud

Use the attached files to test the exercise.

The Tyrion Class MUST inherit Lannister;  
Its constructor MUST call the one from the parent.

If Everything is OK, give all the points.

Yes

No

### Exercise 01: Words of honor

Exercise 01: Words of honor

Use the attached files to test the exercise.

test1.php must work properly.

test2.php must end with a fatal error.

The submitted Class should simply contain protected attributes.

If Everything is OK, give all the points.

Yes

No

### Exercise 02: Fireproofing

Exercise 02: Fireproofing

Use the attached files to test the exercise.

The submitted Class should contain a getBurned method which calls static::resistsFire. However there is some other way to do it discuss it.

If Everything is OK, give all the points.

Yes

No

### Exercise 03: Playing house

Exercise 03: Playing house

Use the attached files to test the exercise.

test1.php must work properly.

test2.php must end with a fatal error.

The submitted Class must have 3 abstract methods (getHouse{Name,Motto,Seat}) and a introduce method which calls the 3 methods via dynamic linkage.

If Everything is OK, give all the points.

Yes

No

### Exercise 04: His sister ? Seriously ?

Exercise 04: His sister ? Seriously ?

Use the attached files to test the exercise.

The submitted Classes should use the instanceof operator to check the type of the sleepWith argument.

If Everything is OK, give all the points.

Yes

No

### Exercise 05: Winter is coming

### Exercice 05: Winter is coming

Use the attached files to test the exercise.  
test1.php must work properly.  
test2.php must end with a fatal error.

There must be a IFighter interface with a fight() method, as well as a NightsWatch Class which maintains an internal array filled using recruit(), the fight() method checks that everybody member in the array implements the IFighter interface before doing anything.

If Everything is OK, give all the points.

Yes

No

### Exercice 06: The wrong kind of pact

#### Exercice 06: The wrong kind of pact

Use the attached files to test the exercise.  
test1.php must work properly.  
test2.php must end with a fatal error.

There is a lot of ways to solve the Factory problem, the easiest would be to have a copy/clone method in the Fighter Class which return a new instance of the type of the object.

If Everything is OK, give all the points.

Yes

No

## Ratings

Don't forget to check the flag corresponding to the defense

Ok

Empty work

Incomplete work

No author file

Invalid compilation

Norme

Cheat

Crash