



# C1 - Pool PHP

---

C-COD-150

## Day 04

---

In the depths of the OOP !

# Day 04

repository name: pool\_php\_d04  
repository rights: ramassage-tek



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

- Introduction
- Exercise 01 (3pts)
- Exercise 02 (2pts)
- Exercise 03 (3pts)
- Exercise 04 (4pts)
- Exercise 05 (2pts)
- Exercise 06 (5pts)
- Exercise 07 (1pts)



## + INTRODUCTION

---

Today we will continue to go deeper into OOP, you will keep using all the notions from yesterday and

You will also discover a few new things:

- static keyword
- namespace

Except specified otherwise, all messages must be followed by a newline.



## + EXERCISE 01 (3PTS)

Turn in: pool\_php\_d04/ex\_01/Mars.php

Restrictions: None.

Create a new class `Mars`: it will have an attribute `id` for which you will create a getter (`getId`) but no setter.

You must create your class so that the first instance of `Mars` has an `id` of 0, the second an `id` of 1...



Static attributes

Example :

```
Terminal
~/C-COD-150> cat example.php
<?php

include_once("Mars.php");

$rocks = new Mars();
$lite = new Mars();
$snack = new Mars();

echo $rocks->getId() . "\n";
echo $lite->getId() . "\n";
echo $snack->getId() . "\n";
```

```
Terminal
~/C-COD-150> php example.php
0
1
2
~/C-COD-150>
```



## + EXERCISE 02 (2PTS)

---

Turn in: pool\_php\_d04/ex\_02/Astronaut.php

Restrictions: None.

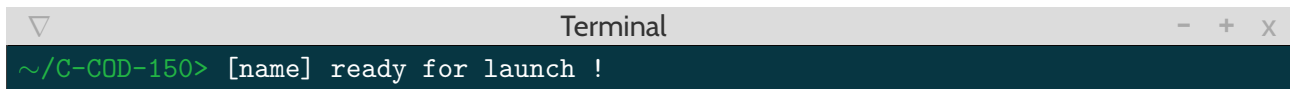
Create a new class `Astronaut` with the following attributes:

- name: a string describing the name of the Astronaut
- snacks: an integer describing the number of snacks our Astronaut possesses
- destination: will be used to store his destination later
- id: an integer describing the id of the Astronaut

The name must be passed during the creation of the Astronaut, it is mandatory, his snack will be initialized to 0 and his destination to null.

The id must be unique and be incremented for each new Astronaut being created (i.e the first Astronaut of our program must have id 0, then the second one will have id 1 etc...)

Moreover, every Astronaut being created will display:



```
Terminal
~/C-COD-150> [name] ready for launch !
```

Where `[name]` is the name of the Astronaut.

You must also add a getter for id (`getId`) but no setter, and a getter for destination (`getDestination`) without setter.

Example :

```
Terminal
~/C-COD-150> cat example.php
<?php

include_once("Astronaut.php");

$mutta = new Astronaut("Mutta");
$hibito = new Astronaut("Hibito");

echo $mutta->getId() . "\n";
echo $hibito->getId() . "\n";
```

```
Terminal
~/C-COD-150> php example.php
Mutta ready for launch !
Hibito ready for launch !
0
1
~/C-COD-150>
```



### + EXERCISE 03 (3PTS)

---

Turn in: pool\_php\_d04/ex\_03/Mars.php

Restrictions: None.

Use your class `Mars` from the first exercise without changing it.

You must create another class `Mars` representing the planet.

To differentiate the two `Mars`, you'll put the first one (the one from exercise 1) in a namespace called `chocolate` and the second one in a namespace called `planet`.



You should turn in only one file `Mars.php`, all your code must be in this file.

You will also add a private attribute `size` to this new `Mars` class, with a getter and a setter (`getSize` and `setSize`).

It must be possible to specify the size of the planet when creating one thanks to its constructor, but it shouldn't be mandatory.



## + EXERCISE 04 (4PTS)

Turn in: pool\_php\_d04/ex\_04/Astronaut.php

pool\_php\_d04/ex\_04/Mars.php

**Restrictions:** In this exercise every time we ask for [Name] to be displayed it should be replaced by the name of the Astronaut for example:

```
Terminal
~/C-COD-150> [Name]: Nothing to do.
```

Will in the end becomes:

```
Terminal
~/C-COD-150> Mutta: Nothing to do.
```

If our Astronaut is named `Mutta`.

It is time for our Astronaut to start working!

If it is not already done, add a getter for his snacks (`getSnacks`), no setter should be created.

Create a new method `doActions` taking an optional parameter.

If no parameter is given the method simply display:

```
Terminal
~/C-COD-150> [Name]: Nothing to do.
```

If the parameter is a `planet::Mars` you will display:

```
Terminal
~/C-COD-150> [Name]: started a mission !
```

Because some people can't guess obvious things (Obviously I won't say who...), you will also need to store our Astronaut's new destination to his attribute `destination` (see Exercise 2).

On the contrary if a `chocolate::Mars` is given you will display:

```
Terminal
~/C-COD-150> [Name] is eating mars number [Mars id].
```

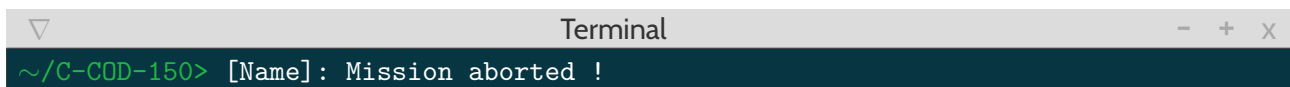




Where [Mars id] should obviously be replaced by the Mars id.

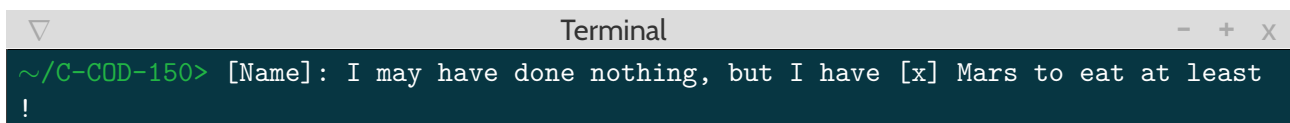
Moreover, if our Astronaut received a snack, you will also need to increment his attribute `snacks` by one.

Finally, during its destruction, if a destination has been selected it will display:

A terminal window titled "Terminal" with a dropdown arrow on the left and window control buttons on the right. The prompt is ~/C-COD-150> and the output is [Name]: Mission aborted !

```
~/C-COD-150> [Name]: Mission aborted !
```

If no destination was selected it will display:

A terminal window titled "Terminal" with a dropdown arrow on the left and window control buttons on the right. The prompt is ~/C-COD-150> and the output is [Name]: I may have done nothing, but I have [x] Mars to eat at least !

```
~/C-COD-150> [Name]: I may have done nothing, but I have [x] Mars to eat at least !
```

Where [x] should be replaced by the number of snacks possessed by our Astronaut.



## + EXERCISE 05 (2PTS)

---

Turn in: pool\_php\_d04/ex\_05/Phobos.php

Turn in: pool\_php\_d04/ex\_05/Mars.php

Restrictions: None.

Use your classes `Mars` from previous exercises without changing it.

You must create a class `Phobos` in the file `Phobos.php`.

This class must be in a namespace `moon` itself defined in the namespace `planet`.

Your class `Phobos` will have a private attribute named `mars` with a getter (`getMars`) but no setter.

This attribute must be specified at the creation, it will be a representation of a planet::`Mars`.

During its creation it will display if it correctly received a planet::`Mars`:

```
Terminal
~/C-COD-150> Phobos placed in orbit.
```

Otherwise it will display:

```
Terminal
~/C-COD-150> No planet given.
```

Example :

```
Terminal
~/C-COD-150> cat example.php
<?php

include_once("Phobos.php");

$titi = new planet\Mars();
$toto = new planet\Mars();

new planet\moon\Phobos($titi);
new planet\moon\Phobos($toto);
```



```
Terminal
~/C-COD-150> php example.php
Phobos placed in orbit.
Phobos placed in orbit.
~/C-COD-150>
```



## + EXERCISE 06 (5PTS)

Turn in: pool\_php\_d04/ex\_06/Phobos.php

Turn in: pool\_php\_d04/ex\_06/Mars.php

Turn in: pool\_php\_d04/ex\_06/Astronaut.php

Turn in: pool\_php\_d04/ex\_06/Team.php

Restrictions: None.

Create a new class `Team` that will represent a team of astronauts.

This class must take a name as parameter during its construction that will be the name of team.

You will create an array which will contain all of your Astronauts and a getter (`getAstronauts`).

You will also create a getter (`getName`) but no setter for this attribute.

You will also create a few methods to manipulate our team.

The first one shall be `add`, it will take a parameter that shall be equal to an Astronaut.

If the parameter given is not an astronaut you must display:

```
Terminal
~/C-COD-150> [Team name]: Sorry, you are not part of the team.
```

Otherwise, you shall display nothing but the Astronaut should be added to the team.

You will also implement a method `remove` that will take an astronaut as parameter and remove him from the team.

This function will never display anything.

You will create a method `countMembers` that will return the number of Astronauts currently in our team.

When an astronaut is going to a destination, he is on a mission.

Finally you will create a method `showMembers` that will display the members contained in the team like this:

```
Terminal
~/C-COD-150> [Team name]: [Astronaut 1] on mission, [Astronaut 2] on standby.
```

Obviously, you will display `on mission` if the Astronaut is currently on mission, otherwise you will display `on standby`.

If there are no member in the team, it will display nothing.



Example :

```
Terminal
~/C-COD-150> cat example.php
<?php

include_once("Team.php");
include_once("Mars.php");

$mutta = new Astronaut("Mutta");
$hibito = new Astronaut("Hibito");
$serika = new Astronaut("Serika");
$spaceBro = new Team ("SpaceBrothers");

$spaceBro->add($mutta);
$spaceBro->add($hibito);
$spaceBro->add($serika);
$spaceBro->add(3);

echo $spaceBro->countMembers() . "\n";

$titi = new planet\Mars();

$mutta->doActions($titi);
$spaceBro->showMembers();
$spaceBro->remove($hibito);

echo $spaceBro->countMembers() . "\n";
```

```
Terminal
~/C-COD-150> php example.php
Mutta ready for launch !
Hibito ready for launch !
Serika ready for launch !
SpaceBrothers: Sorry, you are not part of the team.
3
Mutta: started a mission !
SpaceBrothers: Mutta on mission, Hibito on standby, Serika on standby.
2
Serika: I may have done nothing, but I have 0 Mars to eat at least !
Hibito: I may have done nothing, but I have 0 Mars to eat at least !
Mutta: Mission aborted !
~/C-COD-150>
```



## + EXERCISE 07 (1PTS)

---

Turn in: pool\_php\_d04/ex\_07/Phobos.php

Turn in: pool\_php\_d04/ex\_07/Mars.php

Turn in: pool\_php\_d04/ex\_07/Astronaut.php

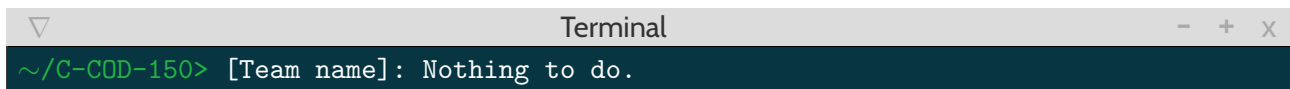
Turn in: pool\_php\_d04/ex\_07/Team.php

Restrictions: None.

Copy your previous exercise files.

Add a new method to your Team, `doActions`, the goal of this method is to call every `doActions` of the Team's Astronauts with the parameter received.

If no parameter is received, you will display:



```
Terminal
~/C-COD-150> [Team name]: Nothing to do.
```

Only once.

In the case of a chocolate received, we will admit that they share the chocolate between themselves but still count it as a full chocolate.

Moreover, you will need to modify your Astronaut class, now that our Astronauts have more experience, they can also go on a mission to Phobos.