

Lab exercise 1

- Install phpunit into your linux home directory and check that it's working successfully.

Lab exercise 2

- Write a factorial function, `factorial($n)`
- Tests:
 - `factorial(0)` should be 1.
 - `factorial(1)` should be 1.
 - `factorial(5)` should be 120.
 - Factorial of any random int, $n > 3$ should be equal to factorial of $n-1$ multiplied by n .
 - Factorial of -3 should be null.
 - Factorial of 1.5 should be null.
 - Factorial of false should be null.
 - Factorial of 'abc' should be null.

```
$ wget -O phpunit https://phar.phpunit.de/phpunit-9.phar
```

```
$ chmod +x phpunit
```

```
$ ll | grep phpunit
```

```
flerro98@ubuntu:~$ wget -O phpunit https://phar.phpunit.de/phpunit-9.phar
--2022-03-27 06:34:18-- https://phar.phpunit.de/phpunit-9.phar
Resolving phar.phpunit.de (phar.phpunit.de)... 188.94.27.25
Connecting to phar.phpunit.de (phar.phpunit.de)|188.94.27.25|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://phar.phpunit.de/phpunit-9.5.19.phar [following]
--2022-03-27 06:34:19-- https://phar.phpunit.de/phpunit-9.5.19.phar
Reusing existing connection to phar.phpunit.de:443.
HTTP request sent, awaiting response... 200 OK
Length: 4525141 (4.3M) [application/octet-stream]
Saving to: 'phpunit'

phpunit          100%[=====] 4.32M  238KB/s  in 18s
2022-03-27 06:34:37 (239 KB/s) - 'phpunit' saved [4525141/4525141]

flerro98@ubuntu:~$ chmod +x phpunit
flerro98@ubuntu:~$ ll | grep phpunit
-rwxrwxr-x 1 flerro98 flerro98 4525141 Mar 15 02:58 phpunit*
```

```
$ sudo apt install php7 php7-mbstring php7-dom php-xml
```

```
(or) $ sudo apt install php-mbstring php-dom php-xml
```

```
$ ./phpunit --version
```

```
flerro98@ubuntu:~$ ./phpunit --version
PHPUnit 9.5.19 #StandWithUkraine
```

```
$ mkdir -p Desktop/lab1/tests/unit
```

```
$ gedit Desktop/lab1/phpunit.xml &
```

```
<phpunit
bootstrap="bootstrap.php">
<testsuites>
  <testsuite name="Sample">
    <directory>tests</directory>
  </testsuite>
</testsuites>
</phpunit>
```

```
Open  ▼  [icon]  phpunit.xml  Save
~/Desktop/lab1

1 <!--
2 * Unit & Automated Testing - Lab 1
3 * Author: Mahmoud Mohamed Kamal
4 * Date: 27-03-1998
5 * File: phpunit.xml
6 -->
7
8 <phpunit bootstrap="bootstrap.php">
9   <testsuites>
10     <testsuite name="Sample">
11       <directory>tests</directory>
12     </testsuite>
13   </testsuites>
14 </phpunit>
```

```
$ gedit Desktop/lab1/bootstrap.php &
```

```
<?php
// my app includes
include('./factorial.php');
```

```
Open  ▼  [icon]  bootstrap  Save
~/Desktop/

bootstrap.php ×  SampleTest.php ×

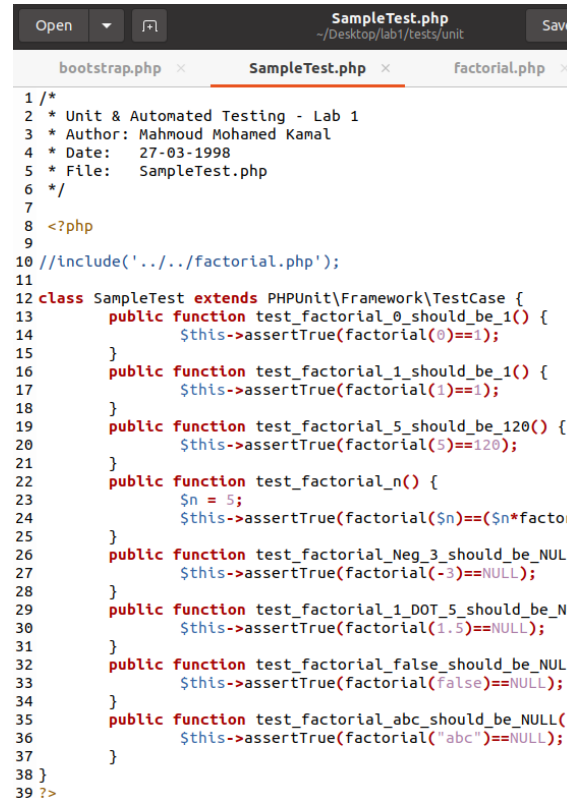
1 /*
2 * Unit & Automated Testing - Lab 1
3 * Author: Mahmoud Mohamed Kamal
4 * Date: 27-03-1998
5 * File: bootstrap.php
6 */
7
8 <?php
9 // my app includes
10 include('./factorial.php');
```

\$ gedit Desktop/tests/unit/SampleTest.php &

<?php

```
//include('../factorial.php');
```

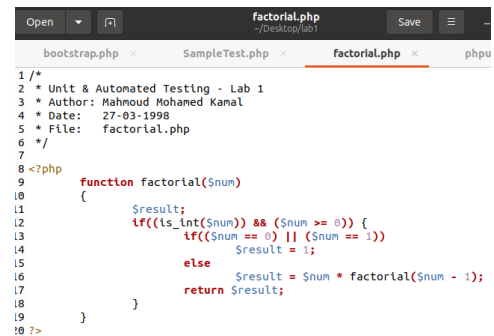
```
class SampleTest extends PHPUnit\Framework\TestCase {
    public function test_factorial_0_should_be_1() {
        $this->assertTrue(factorial(0)==1);
    }
    public function test_factorial_1_should_be_1() {
        $this->assertTrue(factorial(1)==1);
    }
    public function test_factorial_5_should_be_120() {
        $this->assertTrue(factorial(5)==120);
    }
    public function test_factorial_n() {
        $n = 5;
        $this->assertTrue(factorial($n)==($n*factorial($n-1)));
    }
    public function test_factorial_Neg_3_should_be_NULL() {
        $this->assertTrue(factorial(-3)==NULL);
    }
    public function test_factorial_1_DOT_5_should_be_NULL() {
        $this->assertTrue(factorial(1.5)==NULL);
    }
    public function test_factorial_false_should_be_NULL() {
        $this->assertTrue(factorial(false)==NULL);
    }
    public function test_factorial_abc_should_be_NULL() {
        $this->assertTrue(factorial("abc")==NULL);
    }
}
?>
```

A screenshot of a code editor window titled 'SampleTest.php' with the path '~/Desktop/lab1/tests/unit'. The editor shows the PHP code for the SampleTest class, which extends PHPUnit\Framework\TestCase. It includes several test methods: test_factorial_0_should_be_1, test_factorial_1_should_be_1, test_factorial_5_should_be_120, test_factorial_n, test_factorial_Neg_3_should_be_NULL, test_factorial_1_DOT_5_should_be_NULL, test_factorial_false_should_be_NULL, and test_factorial_abc_should_be_NULL. The code uses \$this->assertTrue for assertions. The editor has tabs for 'bootstrap.php', 'SampleTest.php', and 'factorial.php'.

\$ gedit Desktop/lab1/factorial.php &

<?php

```
function factorial($num)
{
    $result;
    if((is_int($num)) && ($num >= 0)) {
        if(($num == 0) || ($num == 1))
            $result = 1;
        else
            $result = $num * factorial($num - 1);
    }
    return $result;
}
?>
```

A screenshot of a code editor window titled 'factorial.php' with the path '~/Desktop/lab1'. The editor shows the PHP code for the factorial function. It uses a recursive approach with a base case for 0 and 1. The code is wrapped in a function definition. The editor has tabs for 'bootstrap.php', 'SampleTest.php', 'factorial.php', and 'phpu'.

\$ ~/phpunit --testdox

```
/*
 * Unit & Automated Testing - Lab 1
 * Author: Mahmoud Mohamed Kamal
 * Date: 27-03-1998
 * File: SampleTest.php
 */

PHPUnit 9.5.19 #StandWithUkraine

Sample
✓ Factorial 0 should be 1
✓ Factorial 1 should be 1
✓ Factorial 5 should be 120
✓ Factorial n
✓ Factorial Neg 3 should be NULL
✓ Factorial 1 DOT 5 should be NULL
✓ Factorial false should be NULL
✓ Factorial abc should be NULL

Time: 00:00.009, Memory: 18.00 MB

OK (8 tests, 8 assertions)
```