# Lab: Decorators

This document defines the exercises for the ["Python OOP" course at @Software University.](https://softuni.bg/modules/74/python-advanced)

Please, submit your source code solutions for the described problems to the [Judge System](https://judge.softuni.org/Contests/1946/Decorators-Lab).

## Number Increment

Having the following code:

**def** number\_increment(numbers):

**def** increase():

*#* ***TODO: Implement***

**return**increase()

Complete the code, so it works as expected.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| print(number\_increment([1, 2, 3])) | [2, 3, 4] |

## Vowel Filter

Having the following code:

**def** vowel\_filter(function):

**def** wrapper():

*#* ***TODO: Implement***

**return** wrapper

Complete the code, so it works as expected:

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| @vowel\_filter  def get\_letters():  return ["a", "b", "c", "d", "e"]  print(get\_letters()) | ["a", "e"] |

## Even Numbers

Having the following code:

**def** even\_numbers(function):

**def** wrapper(numbers):

*#* ***TODO: Implement***

**return** wrapper

Complete the code, so it works as expected.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| @even\_numbers  def get\_numbers(numbers):  return numbers  print(get\_numbers([1, 2, 3, 4, 5])) | [2, 4] |

## Multiply

Having the following code:

**def** multiply(times):

**def** decorator(function):

*#* ***TODO: Implement***

**return** decorator

Complete the code, so it works as expected.

### Examples

|  |  |  |
| --- | --- | --- |
| **Test Code** | **Output** | **Comment** |
| @multiply(3)  def add\_ten(number):  return number + 10  print(add\_ten(3)) | 39 | First, we add 3 to 10 = 13, and then we multiply the result by 3: 13 \* 3 = 39 |
| @multiply(5)  def add\_ten(number):  return number + 10  print(add\_ten(6)) | 80 | (6 + 10) \* 5 = 80 |