

# Lab: Iterators and Generators

This document defines the exercises for the ["Python OOP" course at @Software University](#).

Please, submit your source code solutions for the described problems to the [Judge System](#).

## 1. Custom Range

Create a class called **custom\_range** that receives a **start** (int) and an **end** (int) upon initialization. Implement the **\_\_iter\_\_** and **\_\_next\_\_** methods, so the iterator returns the numbers from the start to the end (**inclusive**).

**Note: Submit only the class in the judge system**

### Examples

| Test Code   | Output  |
|---|---|
| <pre>one_to_ten = custom_range(1, 10) for num in one_to_ten:     print(num)</pre> | 1<br>2<br>3<br>4<br>5<br>6<br>7<br>8<br>9<br>10 |

## 2. Reverse Iter

Create a class called **reverse\_iter** which should receive an iterable upon initialization. Implement the **\_\_iter\_\_** and **\_\_next\_\_** methods, so the iterator returns the items of the iterable in **reversed** order.

**Note: Submit only the class in the judge system**

### Examples

| Test Code  | Output           |
|--|------------------|
| <pre>reversed_list = reverse_iter([1, 2, 3, 4]) for item in reversed_list:     print(item)</pre> | 4<br>3<br>2<br>1 |

## 3. Vowels

Create a class called **vowels**, which should receive a string. Implement the **\_\_iter\_\_** and **\_\_next\_\_** methods, so the iterator returns only the vowels from the string.

**Note: Submit only the class in the judge system**

## Examples

| Test Code  | Output                     |
|--|----------------------------|
| <pre>my_string = vowels('Abcedifuty0o') for char in my_string:     print(char)</pre> | A<br>e<br>i<br>u<br>y<br>o |

## 4. Squares

Create a generator function called **squares** that should receive a number **n**. It should generate the squares of all numbers from **1 to n** (inclusive).

**Note:** Submit only the function in the judge system

## Examples

| Test Code                          | Output            |
|------------------------------------|-------------------|
| <pre>print(list(squares(5)))</pre> | [1, 4, 9, 16, 25] |

## 5. Generator Range

Create a generator function called **genrange** that receives a **start** (int) and an **end** (int) upon initialization. It should generate all the numbers from the **start** to the **end** (inclusive).

**Note:** Submit only the function in the judge system

## Examples

| Test Code                               | Output                          |
|---|---------------------------------|
| <pre>print(list(genrange(1, 10)))</pre> | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] |

## 6. Reverse String

Create a generator function called **reverse\_text** that receives a **string** and yields all string characters on one line in reversed order.

**Note:** Submit only the function in the judge system

## Examples

| Test Code  | Output |
|--|--------|
| <pre>for char in reverse_text("step"):     print(char, end='')</pre> | pets   |