# Lab: Inheritance

Problems for in-class lab for the [Python OOP Course @SoftUni](https://softuni.bg/courses/python-oop). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1940>

# Part I: Inheritance

## Single Inheritance

In a folder called **project** create two files: **animal.py** and **dog.py**.

In the **animal.py** file create a class called Animal with a single public method eat() that returns: **"eating…"**. In the **dog.py** file create a class called Dog with a single public method bark() that returns: **"barking…"**.

Dog should inherit from Animal.

Submit in Judge a **zip file** of thefolder **project**.

## Multiple Inheritance

In a folder called **project** create three files: **person.py** and **employee.py** and **teacher.py**.

In each file create its corresponding class - Person, Employee and Teacher.

Person with a single public method sleep() that returns: **"sleeping..."**

Employee with a single public method get\_fired() that returns: **"fired..."**

Teacher with a single public method teach() that returns: **"teaching..."**

Teacher should inherit from Person and Employee.

Submit in Judge a **zip file** of thefolder **project**.

## Hierarchical Inheritance

In a folder called **project** create three files: **animal.py** and **dog.py** and **cat.py**.

In each file create its corresponding class - Animal, Dog and Cat.

Animal with a single public method eat() that returns: **"eating..."**

Dog with a single public method bark() that returns: **"barking..."**

Cat with a single public method meow() that returns: **"meowing..."**

Dog and Cat should inherit from Animal.

Submit in Judge a **zip file** of thefolder **project**.

# Part II: Reusing Classes

## Random List

Create a RandomList class that has all the functionality of a List.

Add additional function that **returns** and **removes** a random element from the list.

* Public method: get\_random\_element()

## Stack of Strings

Create a class Stack which can store **only strings** and has the following functionality:

* Public field: **data: list**
* Public method: push(item)
* Public method: pop()
* Public method: peek()
* Public method: is\_empty(): returns boolean

Override the string method to return the stack data.