# Lab: Inheritance

This document defines the lab for the ["Java Advanced" course @ Software University](https://softuni.bg/modules/59/java-advanced). Please submit your solutions (source code) of all below described problems in [Judge](https://judge.softuni.bg/Contests/1574/Inheritance-Lab).

<https://softuni.bg/trainings/resources/video/69799/video-28-february-2022-dimo-georgiev-java-oop-february-2022/3587>

# Part I: Inheritance

## Single Inheritance

Create two classes named Animal and Dog.

Animal with a single public method eat() that prints: **"eating…"**

Dog with a single public method bark() that prints: **"barking…"**

TheDog should inherit from Animal.





### Hints

Use the **extends** keyword to build a hierarchy.

## Multiple Inheritance

Create three classes named Animal, Dog, and Puppy.

Animal with a single public method eat() that prints: **"eating…"**

Dog with a single public method bark() that prints: **"barking…"**

Puppy with a single public method weep() that prints: **"weeping…"**

The Dog should inherit from Animal. The Puppy should inherit from Dog.





## Hierarchical Inheritance

Create three classes named Animal, Dog, and Cat.

Animal with a single public method eat() that prints: **"eating…"**

Dog with a single public method bark() that prints: **"barking…"**

Cat with a single public method meow() that prints: **"meowing…"**

Dog and Cat should inherit from Animal.





# Part II: Reusing Classes

## Random Array List

Create a RandomArrayList class that has all the functionality of an ArrayList.

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

* Public method: getRandomElement(): Object

## Stack of Strings

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

* Private field: **data: ArrayList<String>**
* Public method: push(String item): void
* Public method: pop(): String
* Public method: peek(): String
* Public method: isEmpty(): boolean



### Hints

Use composition/delegation to have a field in which to store the stack's data.