# Exercise: Unit Testing

Problems for exercise and homework 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/1949>

## List

You are provided with a simple class - **IntegerList**. It should **only store integers**. **The initial integers should be set by constructor**. They are stored **as a list**. **IntegerList** has a functionality to **add**, **remove\_index** and **get**. Your task is to **test the class**.

***Note: You are not allowed to change the structure of the provided code***

### Constraints

* **add** operation, should **add an element** and returns the list.
  + If the element is not an integer, a **ValueError** is thrown
* **remove\_index** operation removes the element on that index and returns it.
  + If the index is out of range, an **IndexError** is thrown
* **\_\_init\_\_** should only take integers, and store them
* **get** should return the specific element
  + If the index is out of range, an **IndexError** is thrown

### Hint

Do not forget to **test the constructor**

## Extended List

You already have a class - **IntegerList**. We have **modified it** and added some **more functionality** to it. It supports **inserting, getting the biggest number and getting the index of an element**.

***Note: You are not allowed to change the structure of the provided code***

### Constraints

Database should have methods:

* **insert**
  + If the index is out of range, **IndexError** is thrown
  + If the element is not an integer, **ValueError** is thrown
* **get\_biggest**
* **get\_index**

### Hint

Do not forget to test the constructors. They are methods too!

## Car Manager

You are provided with a simple project **containing only one class** - **Car**. The provided class is simple - its **main point is to represent some of the functionality of a Car**. **Each car contains information** about its **make**, **model**, **fuel consumption**, **fuel amount** and **fuel capacity**. Also **each Car can add some fuel** to its tank by refueling and can travel distance by **driving**. In order to be driven, our **Car** needs to **have enough fuel**. Everything in the provided skeleton is working perfectly fine and **you mustn't change it**.

Your job now is to **write unit tests on the provided project** and **it's functionality**. You should test exactly **every part** of code inside the **Car** class:

* You should test **the constructor**
* You should test **all the methods** and **validations inside the class**

### Constraints

* Everything in the provided skeleton is working perfectly fine
* You mustn't change anything in the project structure
* You can test both constructors together
* Any part of validation should be tested
* There is no limit on the tests you will write but keep your attention on the main functionality

***Note: You are not allowed to change the structure of the provided code***

*"Brum…Brum…Brum-suuuututututu…"*