# Lab: First Steps in OOP

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/1934>.

## Rhombus of Stars

Create a program that reads a **positive** **integer N** as input and prints on the console a **rhombus** with size **n**:

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | \* |
| 2 | \*  \* \*  \* |
| 3 | \*  \* \*  \* \* \*  \* \*  \* |
| 4 | \*  \* \*  \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* |
| def print\_row(size, star\_count):      for row in range(size - star\_count):          print(" ", end="")      for row in range(1, star\_count):          print("\*", end=" ")      print("\*")  size = int(input())  for star\_count in range(1, size):    print\_row(size, star\_count)  for star\_count in range(size, 0, -1):    print\_row(size, star\_count) | |

## Scope Mess

Fix the code below, so it returns the expected output. Submit the fixed code in the judge system.

x = **"global"  
  
def** outer():  
 x = **"local"  
  
 def** inner():  
 x = **"nonlocal"** print(**"inner:"**, x)  
  
 **def** change\_global():  
 x = **"global: changed!"** print(**"outer:"**, x)  
 inner()  
 print(**"outer:"**, x)  
 change\_global()  
  
print(x)  
outer()  
print(x)

### Examples

|  |  |
| --- | --- |
| **Current Output** | **Expected Output** |
| global  outer: local  inner: nonlocal  outer: local  global | global  outer: local  inner: nonlocal  outer: nonlocal  global: changed! |
| x = "global"  def outer():      x = "local"      def inner():          nonlocal x          x = "nonlocal"          print("inner:", x)      def change\_global():          global x          x = "global: changed!"      print("outer:", x)      inner()      print("outer:", x)      change\_global()  print(x)  outer()  print(x) | |

## Class Book

Create a class called Book. It should have an \_\_init\_\_() method that should receive:

* name: string
* author: string
* pages: **int**

Submit only the class in the judge system.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| book = Book("My Book", "Me", 200)  print(book.name)  print(book.author)  print(book.pages) | My Book  Me  200 |

## Car

Create a class called Car. Upon initialization it should receive a name, model and engine (all strings). Create a method called get\_info() which will return a string in the following format:   
**"**This is {name} {model} with engine {engine}**"**.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| car = Car("Kia", "Rio", "1.3L B3 I4")  print(car.get\_info()) | This is Kia Rio with engine 1.3L B3 I4 |

## Music

Create class named Music that receives title (string), artist (string) and lyrics (string) upon initialization. The class should also have methods print\_info() and play():

* The print\_info() method should return the following: 'This is "{title}" from "{artist}"'
* The play() method should **return** the **lyrics**.

Submit **only the class** in the judge system. **Test your code** with your own examples.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| song = Music("Title", "Artist", "Lyrics")  print(song.print\_info())  print(song.play()) | This is "Title" from "Artist"  Lyrics |