Lab: Polymorphism

Problems for in-class lab for the Python OOP Course @SoftUni. Submit your solutions in the SoftUni judge system at https://judge.softuni.bg/Contests/1942

1. Execute

Create a function called **`execute`** that receives a **function** as first argument and then **all the other arguments**

Return the **result** of the execution of the passed function with that arguments

Submit only the `execute` function in the judge system

Examples

Test Code	Output

`def say_hello(name, my_name):`
`    print(f"Hello, {name}, I am {my_name}")`
 	
`def say_bye(name):`
`    print(f"Bye, {name}")`
 	
`execute(say_hello, "Peter", "George")`
`execute(say_bye, "Peter")`
	`Hello, Peter, I am George`
	`Bye, Peter`

2. Instruments

Create a function called **`play_instrument`** which will receive an instance of an **instrument** and will print it's **`play()`** method

Submit only the `play_instrument` function in the judge system

Examples

Test Code	Output

`class Guitar:`
`    def play(self):`
`        print("playing the guitar")`
 	
`guitar = Guitar()`
`play_instrument(guitar)`
	`playing the guitar`

`class Piano:`
`    def play(self):`
`        print("playing the piano")`
`piano = Piano()`
`play_instrument(piano)`
	`playing the piano`

3. Shapes

Create an abstract class **Shape** with abstract methods **calculate_area** and **calculate_perimeter**

Create classes **Circle** (receives radius upon initialization) and **Rectangle** (receives height and width upon initialization) that implement those methods (returning the result)

The fields of **Circle** and **Rectangle** should be **private**

Submit all the classes and your imports in the judge system

Examples

Test Code	Output

`circle = Circle(5)`
`print(circle.calculate_area())`
`print(circle.calculate_perimeter())`
	`78.53981633974483`
	`31.41592653589793`

`rectangle = Rectangle(10, 20)`
`print(rectangle.calculate_area())`
`print(rectangle.calculate_perimeter())`
	`200`
	`60`