**Exercises – OOP with C#**

**Exercises 1**: First class and method ToString()

Create a C# program that requests three names of people from the user and stores them in an array of objects of type Person. To do this, first create a Person class that has a Name property of type string and override the ToString() method.

End the program by reading people and executing the ToString() method on the screen.

---

**Input**

1. Juan
2. Sara
3. Carlos

**Output**

1. Hello! My name is Juan
2. Hello! My name is Sara
3. Hello! My name is Carlos

**Exercises 2**: Class person, student and teacher

Create a new C # project with three classes plus another class to test the logic in your code. The main classes of the program are the following classes:

* Person
* Student
* Professor

The Student and Teacher classes inherit from the Person class.

The Student class will include a public Study() method that will write I'm studying on the screen.

The Person class must have two public methods Greet() and SetAge(int age) that will assign the age of the person.

The Teacher class will include a public Explain() method that will write I'm explaining on the screen.

Also create a public method ShowAge() in the Student class that writes My age is: x years oldon the screen.

You must create another test class called StudentProfessorTest with a Main method to do the following:

* Create a new Person and make him say hello
* Create a new Student, set an age, say hello, and display her age on the screen.
* Create a new Teacher, set an age, say hello and start the explanation.

#### ----

#### Input



#### Output

1. Hello!
2. Hello!
3. My age is 21 years old
4. I'm studying
5. Hello!
6. I'm explaining

**Exercises 3**: Implement Interface

Create a C# program that implements an IVehiculo interface with two methods, one for Drive of type void and another for Refuel of type bool that has a parameter of type integer with the amount of gasoline to refuel.

Then create a Car class with a builder that receives a parameter with the car's starting gasoline amount and implements the Drive and Refuel methods of the car.

The Drive method will print on the screen that the car is Driving, if the gasoline is greater than 0. The Refuel method will increase the gasoline of the car and return true.

To carry out the tests, create an object of type Car with 0 of gasoline in the Main of the program and ask the user for an amount of gasoline to refuel, finally execute the Drive method of the car.

#### -----

#### Input

1. 50

#### Output

1. Driving

**Exercise 4**: Abstract class

Create a C# program that implements an abstract class Animal that has a Name property of type text and three methods SetName (string name), GetName and Eat. The Eat method will be an abstract method of type void.

You will also need to create a Dog class that implements the above Animal class and the Eat method that says the dog is Eating.

To test the program ask the user for a dog name and create a new Dog type object from the Main of the program, give the Dog object a name, and then execute the GetName and Eat methods.

----

#### Input

1. Tobby

#### Output

1. Tobby
2. Eating