**OOP Test**

*Name:............................................................................................................... Date: 30.08.2016*

*Github account:......................................................*

***PART 1.***

1. Create interface ***Mammal*** with *methods: talk(), walk(),* Create abstract classes ***Person***and ***Animal***which implements ***Mammal***interface with the following properties: name, age and gender. Create accessor methods for these properties. Keep the properties private. Create appropriate constructor for ***Person***and ***Animal***which accept their properties as parameters.

Create ***Dolphin***class which extends ***Animal*** and implements its methods. Create class ***Student*** which extends ***Person*** and implements its methods.

Create instance of ***Dolphin***and demonstrate talk() and walk() methods. Print dolphin’s name and age.

Create instance of ***Student*** and demonstrate talk() and walk() methods. Print student's name, age and gender.

***PART 2.***

1. Create abstract class ***Vehicle*** with the following properties: brand, model and price. Implement getBrand(), getModel() and getPrice() methods. Create abstract methods:

* + *startEngine()*
  + *stopEngine()*
  + *printData() - print all properties of the vehicle.*

2. Create class ***Car*** that extends the abstract class ***Vehicle*** with properties: horsePower, weight and color. Implements the abstract methods of the parent class: StartEngine and StopEngine (Print some text, for example: “The engine of the car is started” and “The engine of the car is stopped”) Create getHorsePower(), getWeight() and getColor methods. Create constructor with appropriate parameters.

3. Create class ***Bus*** which extends Vehicle and implements its abstract methods in similar way like ***Car***. The ***Bus*** has number of seats, disable people seats and total capacity. Create getNumberOfSeats(), getNumberOfDisablePeopleSeats() and getTotalCapacity(). Create constructor with appropriate parameters.

4. Create instance of ***Car***, print its horsepower, and color and then invoke the following methods: startEngine() and printData();

Create instance of ***Bus***, print its number of seats and invoke the following methods: startEngine(), stopEngine() and printData();

***PART 3.***

1. Create ***PropertyManager*** class which implements the following magical methods:

* *\_\_get($name)*
* *\_\_set($name, $value)*
* *\_\_call($method, $arguments = array())*
* *\_\_callStatic($method, $arguments = array())*
* *\_\_isset($name)*
* *\_\_unset($name)*

Create an instance of the ***PropertyManager*** class and demonstrate how these methods work.

*Tip: Create private array of properties which keeps the data.*