PHP OOP Learn By Doing -2

Coding Exercise

This is the User class:

```
class User {
// Your code goes here
}
```

1. Create a new class property with the name of $firstName, $lastName and prevent any code from outside the class from changing the property value by using the appropriate access modifier (private or protected).
2. Create a method to set the $firstName property value.Remember to use the right access modifier (public/private).
3. Create a method to set the $lastName property value.Remember to use the right access modifier (public/private).
4. Now, create a method to return the $firstName value
5. Now, create a method to return the $lastName value
6. Create a method to return fullName
7. Create a new user object with the name of $user1, set its name to 'John Doe' and make it return its name.
8. Write constructor method to take firstName and lastName as input during the instantiation.
    a. Hint (sample code to write constructor method):

```
public function __construct($arg1, $arg2)
{
if($arg1){ $this -> prop1= $arg1; }
if($arg2){ $this -> prop2= $arg2; }
}
```

9. On the above code use proper variable handling function (isset/empty) to write a bulletproof code.
   Execute This Code:
   $user2 = new user();
   $user3 = new user('Mr.','BASIS');

10. Now modify the constructor function and write in a way so that it can run without risking an error
    Hint: Use default value to the arguments

Execute This Code:
$user2 = new user();
$user3 = new user('Mr.','BASIS');

11. You already have covered the tasks below but you can cross check
    a. Add to the class a constructor method to set a value to the $firstName property as soon as the object is created.
    b. Add to the constructor the ability to set the value of the $lastName property (remember that you can pass to a method more than parameter).
    c. Add to the class a getFullName() public method that returns the full name.
    d. Create a new object, $user1, and pass to the constructor the values of the first and last name. The first name is "John" and the last name is "Doe" (you may choose your preferred combination of first and last name).
    e. Get the full name, and echo it to the screen.
12. Write a function (or you can use any existing function) within the class and use the magic constants below within the function body
    __LINE__ to get the line number in which the constant is used.
    __FILE__ to get the full path or the filename in which the constant is used.
    __METHOD__ to get the name of the method in which the constant is used.
    Explain: What the constants do
13. Add to the class a private property with the name of $username.
14. Create a setter method that can set the value of the $username.
15. Follow the code example below:

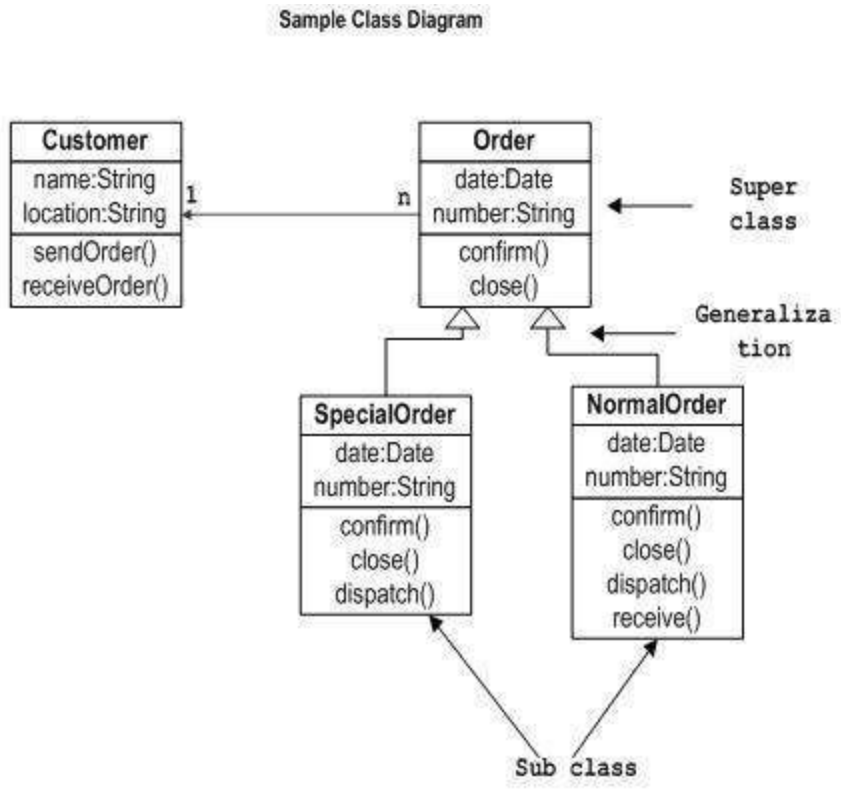
```
class Car { public function hello() {
return "beep"; } }
    // The child class has hello method
    that returns "Hallo"
class SportsCar extends Car { public
function hello() { return "Hallo"; } }
    // Create a new object
$sportsCar1 = new SportsCar();
    // Get the result of the hello
    method
echo $sportsCar1 -> hello();
```

16. Create a class Admin that inherits the User class.
17. Now, let's add to the Admin class some code. First, add a public method by the name of expressYourRole, and make it return the string: "Admin".

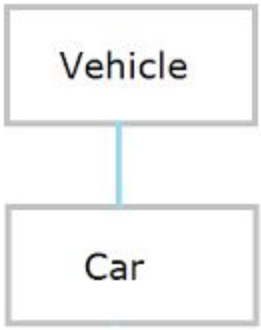
18. Add to the Admin class another public method, sayHello, that returns the string “Hello admin, XXX” with the username instead of XXX.
19. Create an object $admin1 out of the class Admin, set its username to "Balthazar", and say hello to the user. Do you see any problem?
20. What do you think is the cause of the problem?
21. How will you fix this problem?
22. Modify the User Class in a way so that I can run the below code

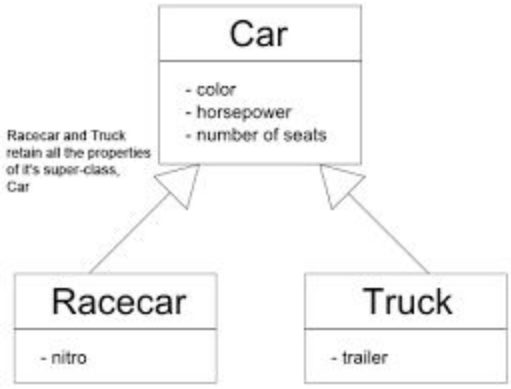
```php
$aUser = new User('Bob', 'Smith', 30);
echo $aUser>getFirstName();
// print 'Bob'
echo $aUser>getLastName();
// prints 'Smith'
echo $aUser->getAge();
// prints '30'
```

23. View the Class Diagram Below. Write the Classes using that. Use your own imagination to any of your queries. Even if you are not right, do not worry. Focus on writing the classes.



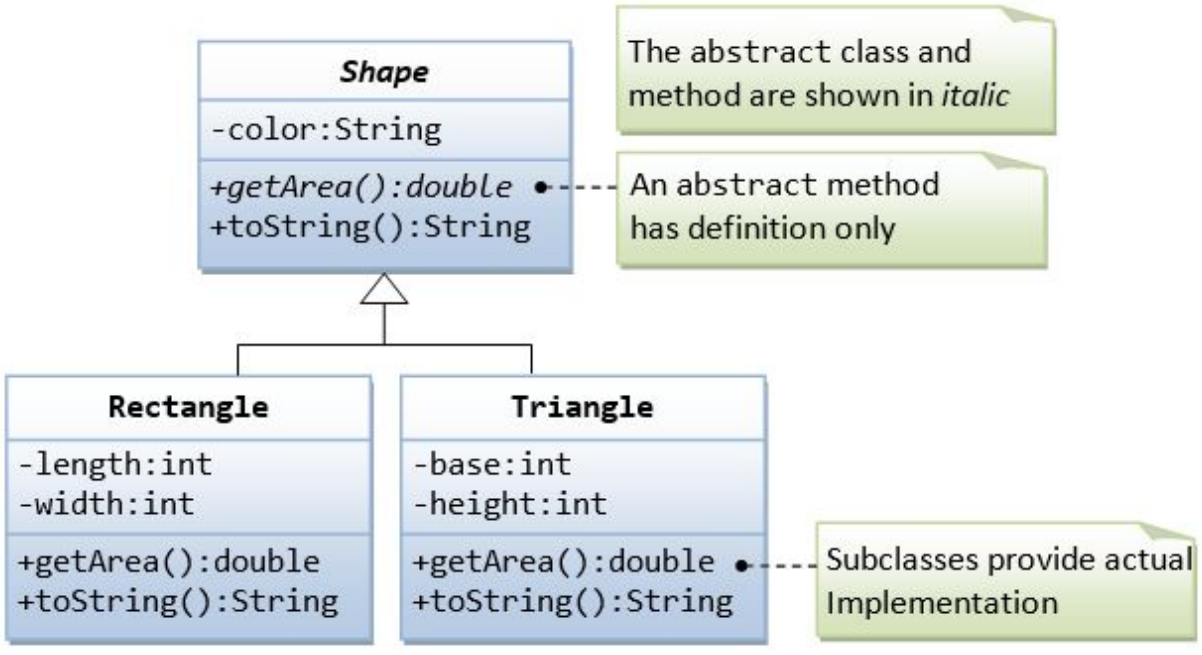
24. See the class diagram below





a. Write classes and subclasses viewing the above class diagram.
b. Introduce drive() method to all classes echo'ing which one is running
c. Run this code $mycar = new Racecar(); $mycar->drive();

25. Analyze the Shape Class Diagram



Update/Create your existing Class : Shape/Rectangle/Triangle/Circle