

FA1 - OOP Concepts in PHP

ICS2609 Application Development and Emerging Technologies 2

Instruction: Perform the following.

1. **Create a class** called <<nameOfClass>> (parent class)
 - a. Include three **member variables** (modifier - private, string type and int type)
 - b. Include a customized **setter** and **getter** methods
 - c. Include **methods** called `displayInfo()` to show the value of the member variables
2. **Create a child class** named <<nameoOfChild>>
 - a. Implement **single inheritance**
 - b. Include another data member (modifier - private, string type)
 - c. Test the inheritance.
 - d. Create a **parameterized constructor** with one argument
 - e. Create a **customized getter method** to retrieve the value of the member variable
 - f. Create a method called `printInfo()` that will display the member variable
3. Modify your source code to implement **multilevel inheritance**.
4. Modify your source code to implement **hierarchical inheritance**.
5. **Override** the `displayInfo()` method. Both shall exists in parent and child class.
6. **Overload** the `printInfo()` method on the <<nameoOfChild>> class. The new method shall have the same name but accepts a single parameter.
7. **Instantiate** three (3) objects and implement the properties and methods.

Note:

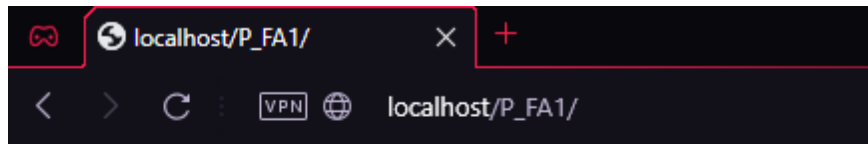
- Take a screenshot for each operation. Make sure that the screenshots are readable.
- Provide expected output for each item if necessary.
- Send a PDF copy your output via BB submission.
- **Do not let your colleagues copy your source codes.**

SCREENSHOTS OF OPERATIONS

1.A-C.

```
index.php x
P_FA1 > index.php
1  <?php
2
3  class Phone{
4      private bool $hasBattery;
5      private string $nameOfOwner;
6      private int $phoneNumber;
7
8      public function displayInfo(){
9          echo "This phone has battery: ". (true === (bool)$this->hasBattery ? 'True' : 'False') . "<br>Owner: " .
10             $this->nameOfOwner . "<br>Phone#:" . $this->phoneNumber;
11      }
12
13      public function setHasBattery($hasBattery){
14          $this->hasBattery = $hasBattery;
15      }
16      public function getHasBattery(){
17          return $this->hasBattery;
18      }
19
20      public function setNameOfOwner($nameOfOwner){
21          $this->nameOfOwner = $nameOfOwner;
22      }
23      public function getNameOfOwner(){
24          return $this->nameOfOwner;
25      }
26
27      public function setPhoneNumber($phoneNumber){
28          $this->phoneNumber = $phoneNumber;
29      }
30      public function getPhoneNumber(){
31          return $this->phoneNumber;
32      }
33  }
34
```

2.A-C. TESTING OF SINGLE INHERITANCE



This phone has battery: True
Owner: John
Phone#:123
OS is open source: False

```
index.php X
P_FA1 > index.php
25
26     public function setPhoneNumber($phoneNumber){
27         $this->phoneNumber = $phoneNumber;
28     }
29     public function getPhoneNumber(){
30         return $this->phoneNumber;
31     }
32 }
33
34 class SailfishOS extends Phone{
35     private bool $isOpenSource;
36
37     public function setIsOpenSource($isOpenSource){
38         $this->isOpenSource = $isOpenSource;
39     }
40     public function getIsOpenSource(){
41         return $this->isOpenSource;
42     }
43 }
44 }
45
46 $phone1 = new SailfishOS();
47 $phone1->setHasBattery(true);
48 $phone1->setNameOfOwner("John");
49 $phone1->setPhoneNumber(123);
50 $phone1->setIsOpenSource(true);
51 $phone1->displayInfo();
52 echo "<br>OS is open source: ". (true === (bool)$phone1->getIsOpenSource() ? 'True' : 'False');
53
54
55
56
57
58
```

2.D-F.

```
index.php X
P_FA1 > index.php
34
35 class SailfishOS extends Phone{
36     private bool $isOpenSource;
37
38     function __construct($parameter){
39         $this->setIsOpenSource($parameter);
40     }
41
42     public function setIsOpenSource($isOpenSource){
43         $this->isOpenSource = $isOpenSource;
44     }
45     public function getIsOpenSource(){
46         return $this->isOpenSource;
47     }
48     public function printInfo(){
49         echo "<br>OS is open source: ". (true === (bool)$this->getIsOpenSource())?'True' : 'False')."";
50     }
51 }
52
53
54 $phone1 = new SailfishOS(true);
55 $phone1->setHasBattery(true);
56 $phone1->setNameOfOwner("John");
57 $phone1->setPhoneNumber(123);
58 $phone1->displayInfo();
59 $phone1->printInfo();
60
```

3-4.

```
53
54 class Android extends Phone {
55
56 }
57
58 class IOS extends Phone {
59
60 }
61
62 class Samsung extends Android{
63
64 }
65
66 class Apple extends IOS{
67
68 }
69
70
71
```

5.

```
index.php X
P_FA1 > index.php
51
52     public function displayInfo(){
53         echo "<br>SAILFISHOS PHONE". "<br>This phone has battery: ". (true === (bool)$this->hasBattery ? 'True' : 'False') .
54         $this->nameOfOwner . " <br>Phone#: " . $this->phoneNumber. " <br>OS is open source: "
55         . (true === (bool)$this->getIsOpenSource()?'True' : 'False')."";
56     }
57 }
58 class Android extends Phone {
59     public function displayInfo(){
60         echo "<br>ANDROID PHONE". "<br>This phone has battery: ". (true === (bool)$this->hasBattery ? 'True' : 'False')
61         . " <br>Owner: " .
62         $this->nameOfOwner . " <br>Phone#: " . $this->phoneNumber;
63     }
64 }
65 class IOS extends Phone {
66     public function displayInfo(){
67         echo "<br>IOS PHONE". "<br>This phone has battery: ". (true === (bool)$this->hasBattery ? 'True' : 'False')
68         . " <br>Owner: " .
69         $this->nameOfOwner . " <br>Phone#: " . $this->phoneNumber;
70     }
71 }
72 class Samsung extends Android{
73     public function displayInfo(){
74         echo "<br>SAMSUNG: ANDROID PHONE". "<br>This phone has battery: ". (true === (bool)$this->hasBattery ? 'True' : 'Fal
75         $this->nameOfOwner . " <br>Phone#: " . $this->phoneNumber;
76     }
77 }
78 class Apple extends IOS{
79     public function displayInfo(){
80         echo "<br>APPLE: IOS PHONE". "<br>This phone has battery: ". (true === (bool)$this->hasBattery ? 'True' : 'False')
81         . " <br>Owner: " .
82         $this->nameOfOwner . " <br>Phone#: " . $this->phoneNumber;
83     }
84 }
85 }
```

6. I could not overload the printInfo() method without utilizing the magic method
"__call"



7. TEST

```
97  $phone1 = new SailfishOS(true);
98  $phone1->setHasBattery(true);
99  $phone1->setNameOfOwner("Jean");
100 $phone1->setPhoneNumber(84311512711);
101 $phone1->displayInfo();
102 $phone1->printInfo();
103
104 $phone2 = new Samsung();
105 $phone2->setHasBattery(true);
106 $phone2->setNameOfOwner("John");
107 $phone2->setPhoneNumber(67348123513);
108 $phone2->setIsOpenSource(true);
109 $phone2->displayInfo();
110
111 $phone3 = new Apple();
112 $phone3->setHasBattery(true);
113 $phone3->setNameOfOwner("Jack");
114 $phone3->setPhoneNumber(81247345681);
115 $phone3->setIsOpenSource(false);
116 $phone3->displayInfo();
117 ?>
```

SAILFISH OS PHONE

This phone has battery: True

Owner: Jean

Phone#:84311512711

OS is open source: True

SAMSUNG: ANDROID PHONE

This phone has battery: True

Owner: John

Phone#:67348123513

OS is open source: True

APPLE: IOS PHONE

This phone has battery: True

Owner: Jack

Phone#:81247345681

OS is open source: False