

Take Home Coding Challenge

Please provide all solution materials in whatever way you see fit.

PART 1:

Given the following output, please provide an appropriate **object oriented definition and implementation**.

Please provide a solution that promotes reusability, and where possible defines appropriate interfaces.

Note that you can (and should) add additional methods and classes as needed.

Please make the solution available to review.

Requirement

```
$foo = new Donor();
$foo->setName('Foo Doe');
$foo->setBirthDate('1990-02-02');
$foo->setGender('m');
$foo->setInterest("Music");
$foo->setInterest("Movies");

$baz = new Owner();
$baz->setName('Baz Doe');
$baz->setBirthDate('1990-02-02');
$baz->setGender('m');

$event1 = new Event("Cool Night Out", "2016-03-11 17:00:00", "30m");
$event2 = new Event("Lunch With Baz", "2016-04-22 12:30:00", "2h");

$bar = new Org();
$bar->setName('Baz Org');
$bar->setAddress('123 E Main St');
$bar->setOrgOwner($baz);
$bar->setOrgId('9494949OR');
$bar->addEvent($event1);
$bar->addEvent($event2);

$foo->attendEvent($event1);

$foo->printJson();
$baz->printJson();
$bar->printJson();
$bar->getNextEvent()->printJson();
```

Output

```
#$ /usr/bin/php mytest.php

{"name":"Foo
Doe","birth_date":"1980-02-02","gender":"m","interests":["Music","Movies"]}

{"name":"Baz Doe","birth_date":"1980-02-02","gender":"m","interests":[]}

{"org_id":"9494949OR","name":"Baz Org","address":"123 E Main
St","owner":"Baz Doe","upcoming_events":[{"name":"Cool Night
Out","start_at":"2016-03-11 17:00:00","duration":"30m"}, {"name":"Lunch With
Baz","start_at":"2016-04-22 12:30:00","duration":"2h"}]}

{"name":"Cool Night Out","starts_at":"2016-03-11
17:00:00","ends_at":"2016-03-11 17:30:00","attendees":[{"name":"Foo
Doe","birth_date":"1980-02-02","gender":"m","interests":["Music","Movies"]
}]}
```

PART 2:

The "setOrgId" method requires a string value that meets a certain pattern.
Please provide a **valid function** that can universally validate the id for the following criteria.
Feel free to integrate the the function into the solution for Part 1.

- The string must be either 9 or 10 characters in length.
- The string must start with either with "94" or "49".
- The string must end with two upper case alpha characters.
- The entire string must be alpha-numeric.

Examples

```
9912345US
9499999XX
4999999YY
94ab55c2CA
```

PART 3:

Please provide a **sample MySQL database schema** to represent the object relationships in your solution.
Please make sure to specify the relationships between tables using a ERD diagram or UML definition.

Also please provide a **SQL query** to produce a report on all attendees for all events for a particular owner.
Please sort the events by start date, and sub sort by attendee name.

The report should have the following headers:
(event_name, 'event_start_date', 'attendee_name', 'attendee_age', 'attendee_gender')

Also provide a **SQL query** to produce a report on event attendance for all events sorted by the event owners name.
The report should have the following headers:
(event_name, 'event_start_date', 'event_owner_name', 'event_owner_id', 'attendee count')

PART 4:

For your implementation to Part 1, describe the process of **testing** your solution.
Feel free to provide code samples of your tests using the PHPUnit Testing Framework.
Be as verbose as possible in describing your process outside of the code itself.