```
 function insertRecords($records_array)
{
        $dogs_array_size = count($this->dogs_array["dog"]);
        for($I=0;$I< count($records_array);$I++)
        {
                $this->dogs_array["dog"][$dogs_array_size + $I] = $records_array[$I];
         }
}
```

■ **Note** The process of creating the `dog_array` using the JSON functions shown previously will create one inconsistency in creating the `dog_array`. If the `dog_data.xml` file contains only one record, the JSON functions will not create a numeric index (such as '0'). When more than one record is contained in the xml file the numeric indexes will be created (such as '0', '1'). An alternative solution to which handles these differences is provided in the demo files on the textbook website.

In the `insertRecords` method, all records are added to the end of the array (the calling program can sort them if needed). The current size of `dogs_array` is determined by the `count` method and stored into `$dogs_array_size`. The `count` method is also used inside the `for` structure to determine the size of the `$records_array` and to determine the number of loops. Since the results of the `count` method produces the size of the array, which is one more than the last subscript position, the result of `count` also gives the next position available to insert a record.

In the first loop, `$I` is 0. The first record of `$records_array` is placed into `$dogs_array_size` plus 0, or `$dogs_array_size` (the first open row to place a record). The next time through the loop, the second record of `$records_array` (`$I` was incremented by the loop) is placed into position `$dogs_array_size` plus 1. This is the next position available after the first record has been inserted. The loop will continue until there are no more records in the `$records_array`. By the way, this method also works well with just one record to insert (as long as it is passed as an associate array). The loop will execute only once.

The last method you need to examine is an `update` method. This method is a very simple form of the destructor method.

```
function updateRecords($records_array)
{
        foreach ($records_array as $records=>$records_value) {
                foreach ($records_value as $record => $record_value) {
                         $this->dogs_array["dog"][$records] = $records_array[$records];
                }
        }
}
```

This little tiny method will take any size associate array and update the dogs array. It is based on PHP's ability to dynamically build arrays.

```
$records_array = Array (
0 => Array ( "dog_name" => "Jeffrey", "dog_weight" => "19", "dog_color" => "Green",
"dog_breed" => "Lab" ),
2 => Array ( "dog_name" => "James", "dog_weight" => "21", "dog_color" => "Black",
"dog_breed" => "Mixed" ));
```