College of Engineering, Construction & Living Sciences
Bachelor of Information Technology
IN607: Introductory Application Development Concepts
Level 6, Credits 15
In-Class Activity: PHP Basics 2 Answers

Problem 1:

```php
<?php
    class Cat {
        private $name;
        private $age;
        private $breed;

        public function __construct($name, $age, $breed) {
            $this->name = $name;
            $this->age = $age;
            $this->breed = $breed;
        }

        public function set_name($name) {
            $this->name = $name;
        }

        public function get_name() {
            return $this->name;
        }

        public function set_age($age) {
            $this->age = $age;
        }

        public function get_age() {
          return $this->age;
        }

        public function set_breed($breed) {
            $this->breed = $breed;
```

```php
        }

        public function get_breed() {
          return $this->breed;
        }

        public function __toString() {
          return "My " . $this->breed . " 's name is "
                . $this->name. ". S/he is " . $this->age . " year(s) old.";
        }
    }

    $cat_one = new Cat("Ralph", 12, "Abyssinian");
    $cat_two = new Cat("Leo", 9, "Birman");
    $cat_one->set_name("Fido");
    $cat_one->set_age(10);
    $cat_two->set_breed("American Bobtail");
    echo $cat_one;
    echo $cat_two;
?>
```

Problem 2:

```php
<?php
    class Employee {
        protected $first_name;
        protected $last_name;
        protected $salary;

        public function __construct($first_name, $last_name, $salary) {
            $this->first_name = $first_name;
            $this->last_name = $last_name;
            $this->salary = $salary;
        }

        public function __toString() {
            return $this->first_name . " " . $this->last_name;
        }
    }

    class SoftwareDeveloper extends Employee {
        public function __construct($first_name, $last_name, $salary, $prog_lang) {
            $this->first_name = $first_name;
            $this->last_name = $last_name;
            $this->salary = $salary;
            $this->prog_lang = $prog_lang;
        }
    }

    class AgileCoach extends Employee {
        public function __construct($first_name, $last_name, $salary, $employees) {
            $this->first_name = $first_name;
            $this->last_name = $last_name;
            $this->salary = $salary;
            $this->employees = $employees;
```

```php
            }

        public function add($employee) {
            array_push($this->employees, $employee);
        }

        public function remove($employee) {
            $key = array_search($employee, $this->employees);
            if ($key !== false) {
                unset($this->employees[$key]);
            }
        }

        public function search($employee) {
            $key = array_search($employee, $this->employees);
            if ($key === false) {
                echo $employee . " not found" . "<br>";
            } else {
                echo $employee . " found" . "<br>";
            }
        }

        public function show_all() {
            foreach($this->employees as $employee) {
                echo $employee . "<br>";
            }
        }
    }

    $sft_dev_one = new SoftwareDeveloper("Alfredo", "Boyle", 50000, "C#");
    $sft_dev_two = new SoftwareDeveloper("Malik", "Martin", 55000, "JavaScript");
    $sft_dev_three = new SoftwareDeveloper("Livia", "Martin", 75000, 'Kotlin');
    $agile_coach = new AgileCoach("Lillian", "Cunningham", 100000, array($sft_dev_one, $sft_dev_two));
    $agile_coach->add($sft_dev_three);
    $agile_coach->remove($sft_dev_one);
    $agile_coach->show_all();
    $agile_coach->search($sft_dev_one);
    $agile_coach->search($sft_dev_three);
?>
```

Problem 3:

```php
<?php
class Language {
    public function good_morning() {
        throw new Exception("good_morning not implemented");
    }
}

class Maori extends Language {
    public function good_morning() {
        echo "Morena";
    }
}
```

```php
    class Spanish extends Language {
        public function good_morning() {
            echo "Hola";
        }
    }

    class German extends Language {
        public function good_morning() {
            echo "Guten Morgen";
        }
    }

    $maori = new Maori();
    $spanish = new Spanish();
    $german = new German();
    $maori->good_morning();
    $spanish->good_morning();
    $german->good_morning();
?>
```

Problem 4:

This is an okay solution, but we can limit the size of our **stack**. Also, there is an absence of error checking, in particular, when pushing & popping to & from the **stack**.

```php
<?php
    class Stack {
        private $stack;
        private $limit;

        public function __construct() {
            $this->stack = array();
            $this->limit = 5; // Limit the number of elements to five
        }

        public function push($item) {
            // Okay solution
            array_unshift($this->stack, $item);

            // Better solution
            if (count($this->stack) < $this->limit) {
                array_unshift($this->stack, $item);
            } else {
                throw new RunTimeException("The stack is full. You can not add an item.");
            }
        }

        public function pop() {
            // Okay solution
            return array_shift($this->stack);

            // Better solution
            if ($this->is_empty()) {
                throw new RunTimeException("The stack is empty. You can not remove a item.");
            } else {
                return array_shift($this->stack);
```

```php
            }
        }

        public function peek() {
            return current($this->stack);
        }

        public function is_empty() {
            return empty($this->stack);
        }

        public function size() {
            return count($this->stack);
        }

        public function show_all() {
            foreach($this->stack as $item) {
                echo $item . "<br>";
            }
        }

        public function __toString() {
            return $this->stack;
        }
    }

    $stack = new Stack();
    $stack->push("Introductory App Dev Concepts");
    $stack->push("Intermediate App Dev Concepts");
    $stack->push("Advanced App Dev Concepts");
    $stack->pop();
    $stack->show_all();
    echo $stack->peek() . " is at the top of the stack" . "<br>";
    echo "There are " . $stack->size() . " item(s) in the stack"
?>
```

Problem 5:

```php
<?php
    interface App {
        public function login($email, $password);
        public function register($email, $password, $username);
        public function logout();
    }

    class Facebook implements App {
        public function login($email, $password) {
            echo "Logged in with the email - $email.";
        }

        public function register($email, $password, $username) {
            echo "Registered with the email and username - $email and $username.";
        }

        public function logout() {
```

```php
            echo "User logged out.";
        }
    }

    $fb_user = new Facebook();
    $fb_user->register("john.doe@gmail.com", "P@ssw0rd123", "john.doe") . "<br>";
    $fb_user->login("john.doe@gmail.com", "P@ssw0rd123") . "<br>";
    $fb_user->register() . "<br>";
    $fb_user->logout() . "<br>";
?>
```