Evidence for Implementation and Testing Unit.

Your name here Marta Dabrowka
Your Cohort E16
Date here: 25.01.2018

I.T 1- Demonstrate one example of encapsulation that you have written in a program.

```
public class Ingredient {
   private String type;
   private int number;


    public Ingredient(String type, int number) {
        this.type = type;
        this.number = number;
    }

    public String getType() { return this.type; }

    public int getNumber() { return this.number; }

    public void decreaseNumber() { this.number -= 1; }

    public void increaseNumber(int extraPortions) { this.number += extraPortions; }


}
```

I.T 2 - Example the use of inheritance in a program.

A class:

```
public abstract class MenuItem implements Menuable {
    private String name;
    private double price;


    public MenuItem(String name, double price) {
        this.name = name;
        this.price = price;

    }

    public double getPrice() { return this.price; }

    public String getName() { return this.name; }


}
```

A class that inherits from the previous class:

```java
public class Dish extends MenuItem implements Ingredientable {
    private String name;
    private double price;
    private ArrayList<Ingredient> ingredients;


    public Dish(String name, double price) {
        super(name, price);
        this.ingredients = new ArrayList<Ingredient>();
    }

    public ArrayList getIngredients() {
        return new ArrayList<Ingredient>(ingredients);
    }

    public int ingredientCount() { return ingredients.size(); }


    public void addIngredient(Ingredient ingredient) {
        if (ingredient.getNumber() >= 1)
        ingredients.add(ingredient);
    }


    public void removeIngredient(Ingredient ingredient) { ingredients.remove(ingredient); }


    public boolean checkIfDishContainsIngredient(Ingredient ingredient) {
        return ingredients.contains(ingredient);
    }


    public boolean checkIIngredientIsAvailable() {
         for (Ingredient ingredient: ingredients) {
             if (ingredient.getNumber() >= 1);
             return true;
         }
        return false;
    }
```

An object in the inherited class

```java
dish = new Dish( name: "Ham sandwich",  price: 5);
ingredients = new ArrayList();
ham = new Ingredient( type: "ham",  number: 2);
bread = new Ingredient( type: "bread",  number: 5);
```

A method that uses the information inherited from another class

```java
public double setDiscountPrice() {
    double newPrice = this.price * 0.75;
    return newPrice;
}
```

I.T 3 - Example of searching

(if you do not have a search and sort algorithm, write one up, take a screenshot. Remember to include the results as well.)

```ruby
def self.find(id)
  sql = "SELECT * FROM books WHERE id = $1"
  values = [id]
  results = SqlRunner.run(sql,values)[0]
  book = Book.new(results)
  return book
end
```

```
[3] pry(main)> Book.find(3)
=> #<Book:0x007fd2d0071a48
 @author_id=2,
 @buy_price=8,
 @cover_image=
  "https://static1.squarespace.com/static/5019421184ae7e81bd01bf4e/551ef206e4b0170ec8abdbe0/56bbf4f562cd94f5f8c27ae7/145
5158518149/HA+8-2-16.jpeg",
 @genre_id=1,
 @id=3,
 @quantity=34,
 @sell_price=10,
 @source_language_id=6,
 @title="Human Acts">
```

I.T 4 – Example of sorting

```ruby
fruits = ["pear", "plum", "peach", "passionfruit", "pomelo"]

def fruits_sort(my_array)
 my_array.sort
end

puts fruits_sort(fruits)
```

```
➜ implementation_and_testing ruby array.rb
passionfruit
peach
pear
plum
pomelo
```

I.T 5 - Example of an array, a function that uses an array and the result

array.rb hash_pda.rb

```ruby
fruits = ["pear", "plum", "peach", "passionfruit", "pomelo"]

def fruits_number(my_array)
  my_array.length
end

puts "I have #{fruits_number(fruits)} different fruit."
```

```
➜ implementation_and_testing ruby array.rb
I have 5 different fruit.
```

I.T 6 - Example of a hash, a function that uses a hash and the result

```ruby
fave_dog1 = {
  name: "Lulu",
  breed:  "dachshund",
  colour: "brown",
  yappy: false,
  cuteness: true
}

def dog_evaluation(my_hash)
  if my_hash[:cuteness] == true && my_hash[:yappy] == false
    return "You can be my doggy, #{my_hash[:name]}"
  else
    return "Sorry, I cannot keep you"
  end
end

puts dog_evaluation(fave_dog1)
```

```
➜  implementation_and_testing ruby hash_pda.rb
You can be my doggy, Lulu
```

I.T 7 - Example of polymorphism in a program

```java
public class Collector {
    private String name;
    private ArrayList<Collectible> collectionItems;


    public Collector(String name) {
        this.name = name;
    }


    public String getName() {
        return this.name;
    }

    public void addToCollection(Collectible item) {
        collectionItems.add(item);
    }
}
```

```java
public interface Collectible {

    public double calculateValue();

}
```

```java
public class Vehicle implements Collectible {
    private String companyName;
    private String model;
    private double price;
    private double sellerValue;
    private int age;
    private int numberOfWheels;


    public Vehicle(String companyName, String model, double price, double sellerValue, int age, int numberOfWheels) {
        this.companyName = companyName;
        this.model = model;
        this.price = price;
        this.sellerValue = sellerValue;
        this.age = age;
        this.numberOfWheels = numberOfWheels;
    }



    public double calculateValue() {
        return this.price * this.sellerValue * this.numberOfWheels;
    }
```

```java
public class Painting implements Collectible {
    private String title;
    private String artist;
    private double price;
    private double sellerValue;
    private int age;


    public Painting(String title, String artist, double price, double sellerValue, int age) {
        this.title = title;
        this.artist = artist;
        this.price = price;
        this.sellerValue = sellerValue;
        this.age = age;
    }



    public double calculateValue() {
        return this.age * this.price * this.sellerValue;
    }

    public double getPrice() {
        return this.price;
    }
```