

## Evidence for Implementation and Testing Unit

Mark Conroy

Cohort E17

### I.T 1- Demonstrate one example of encapsulation

Private String name

```
private String name;
private int hieght;
private int length;
private int weight;
private String type;
private int healthValue;

public Dinosaur(String name, int height, int length, int weight, String type, int healthValue) {
    this.name = name;
    this.hieght = height;
    this.length = length;
    this.weight = weight;
    this.type = type;
    this.healthValue = healthValue;
}
```

### I.T 2- Example of inheritance

A Class

```
package dinosaurs;

public abstract class Dinosaur implements IFeed{

    private String name;
    private int hieght;
    private int length;
    private int weight;
    private String type;
    private int healthValue;

    public Dinosaur(String name, int height, int length, int weight, String type, int healthValue) {
        this.name = name;
        this.hieght = height;
        this.length = length;
        this.weight = weight;
        this.type = type;
        this.healthValue = healthValue;
    }
}
```

A class that inherits from the previous class

```
package aquatic_dinosaurs;

import dinosaur_attack.IAttack;
import dinosaurs.*;

import java.util.Random;

public abstract class AquaticDinosaurs extends Dinosaur {

    private int rage;
    private IAttack attack;

    public AquaticDinosaurs(String name, int height, int length, int weight, String type, int healthValue) {
        super(name, height, length, weight, type, healthValue);
        this.rage = 0;
    }

    public void canRage() {
        Random rand = new Random();
        int amountOfRage = rand.nextInt( bound: 100);
        this.rage += amountOfRage;
    }

    public int getRage() { return rage; }
```

An object in the inherited class

```
Tusoteuthis tusoteuthis;
Edible edible;
IAttack swipe;
Dinosaur dinosaur;

@Before
public void before() {
    tusoteuthis = new Tusoteuthis( name: "Simon", height: 500, length: 8, weight: 200, type: "Aquatic", healthValue: 300);
}
```

A method that uses information inherited from another class

```
public void takeDamage(int damage) {
    this.healthValue -= damage;
}

public void feed(int edible){
    this.healthValue += edible;
}
```

### I.T 3- Example of Searching

A function that searches data

```
def self.total_by_tag(tag_id)
  sql = "SELECT sum(value) FROM transactions WHERE tag_id = $1"
  values = [tag_id]
  values = SqlRunner.run(sql, values)
  value = values.first
  return value['sum']
end
```

The result of the function running

## Total By Type

Total Amount By Transaction Type = £2030



#### I.T 4- Example of Sorting

Function that sorts data

```
def sort(languages)
  languages.sort!{|x, y| x <=> y}
end

p sort(languages)
```

The result of the function running

```
→ week_01 git:(master) X ruby pda_array.rb
["Java", "Javascript", "Python", "Ruby"]
→ week_01 git:(master) X █
```

#### I.T 5- Example of an array

An array and a function that uses the array

```
languages = ["Ruby", "Python", "Java", "Javascript"]

def method_name(languages)
  languages.each {|i| puts i.swapcase}
end
```

The result of the function running

```
→ week_01 git:(master) ✗ ruby pda_array.rb
rUBY
pYTHON
jAVA
jAVASCRIPT
["Ruby", "Python", "Java", "Javascript"]
→ week_01 git:(master) ✗
```

## I.T 6- Example of a hash

A hash and a function that uses it

```
jedi = [
  {name: "Luke", Lightsaber: "green", quote:"I am a jedi, like my father before me."},
  {name: "Obi Wan", Lightsaber: "blue", quote:"Well hello there."},
  {name: "Mace Windu", Lightsaber: "Purple", quote:"A SITH LORD!?"}
]

def get_quote(list, name)
  found = list.find_all { |item| item[:name] == name }
  return found.map { |item| item[:quote] }
end

puts get_quote(jedi, "Luke")
```

The result of the function running

```
→ week_01 git:(master) ✗ ruby pda_hash.rb
I am a jedi, like my father before me.
→ week_01 git:(master) ✗
```

## I.T 7- Example of Polymorphism

Interface

```
import dinosaurs_food.Edible;  
  
public interface IFeed {  
    void feed(Edible edible);  
}
```

Interface type

```
package dinosaurs_food;  
  
public interface Edible {  
}
```

Method that uses interface

```
public void feed(int edible){  
    this.healthValue += edible;  
}
```

Class that uses the method

```
package dinosaurs;  
  
public abstract class Dinosaur implements IFeed{
```