**Evidence for Implementation and Testing Unit**

Mark Conroy
Cohort E17

I.T 1- Demonstrate one example of encapsulation

Private String name

```
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;
    }

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

    public int getHeight() {
        return this.hieght;
    }

    public int getLength() {
        return this.length;
    }

    public int getWeight() {
        return this.weight;
    }

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

I.T 2- Example of inheritance

A Class

```java
import dinosaur_attack.IAttack;
import dinosaurs.*;

import java.util.Random;

public abstract class AqaticDinosaurs extends Dinosaur {

    private int rage;
    private IAttack attack;

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

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

    public int getRage() { return rage; }

}
```

A class that inherits from the previous class

```java
import ...

public class Plesiosaur extends AqaticDinosaurs {

    private String discription;
    private IAttack swipe;

    public Plesiosaur(String name, int height, int length, int weight, String type, int helathValue) {
        super(name, height, length, weight, type, helathValue);
        this.discription = discription;
        this.swipe = swipe;
    }

    public String getDiscription() {
        return "Plesiosaurs were an order of large carnivorous marine reptiles from 245 million years ago.";
    }

    public void feed(Edible edible){
        this.feed(edible);
    }

}
```

An object in the inherited class, method that uses information inherited from another class

```java
public class PlesiosaurTest {

    Plesiosaur plesiosaur;
    Edible edible;
    IAttack swipe;
    Dinosaur dinosaur;

    @Before
    public void before() {
        plesiosaur = new Plesiosaur( name: "Nessie",  height: 1,  length: 4,  weight: 450,  type: "Aquatic",  helathValue: 500);
        swipe = new Swipe();
    }

    @Test
    public void plesiosaurCanTakeDamage() {
        plesiosaur.takeDamage(50);
        assertEquals( expected: 450, plesiosaur.getHealth());
    }

    @Test
    public void plesiosaurCanFeed() {
        edible = new Fish();
        plesiosaur.feed( edible: 100);
        assertEquals( expected: 600, plesiosaur.getHealth());
    }

    @Test
    public void plesiosaurCanRage() {
        plesiosaur.canRage();
        assertEquals( expected: true,  actual: plesiosaur.getRage() > 0);
    }
```

I.T 3- Example of Searching

A function that searches data

```ruby
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

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

p sort(languages)
```

The result of the function running

```
➜  week_01 git:(master) ✗ ruby pda_array.rb
["Java", "Javascript", "Python", "Ruby"]
➜  week_01 git:(master) ✗
```

I.T 5- Example of an array

An array and a function that uses the array

```ruby
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

```ruby
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

Class of Shop with an array list of ISell items and a method to populate the array list

```
public class Shop {

    private String name;
    private ArrayList<ISell> stock;

    public Shop(String name) {
        this.name = name;
        this.stock = new ArrayList<ISell>();
    }

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

    public int stockCount() {
        return this.stock.size();
    }

    public void addToStock(ISell item) {
        this.stock.add(item);
    }

    public void removeFromStock(ISell item) {
        this.stock.remove(item);
    }
```

Abstract class of Stock Item that uses ISell

```
public abstract class StockItem implements ISell {

    protected int buyPrice;
    protected int sellPrice;

    public StockItem(int buyPrice, int sellPrice) {
        this.buyPrice = buyPrice;
        this.sellPrice = sellPrice;
    }

    public int getBuyPrice() {
        return this.buyPrice;
    }

    public int getSellPrice() {
        return this.sellPrice;
    }

}
```

Class of Music Book that extends Stock Item

```
public class MusicBook extends StockItem {

    String title;

    public MusicBook(String title, int buyPrice, int sellPrice) {
        super(buyPrice, sellPrice);
        this.title = title;
    }

    public String getTitle() {
        return this.title;
    }

    public int getMarkupPrice() {
        return this.sellPrice -= this.buyPrice;
    }

}
```

Class of Music Stand that extends Stock Item

```
public class MusicStand extends StockItem {

    private String colour;

    public MusicStand(String colour, int buyPrice, int sellPrice) {
        super(buyPrice, sellPrice);
        this.colour = colour;
    }

    public String getColor() {
        return this.colour;
    }

    public int getMarkupPrice() {
        return this.sellPrice -= this.buyPrice;
    }

}
```

The interface of ISell that is used

```
public interface ISell {
    int getMarkupPrice();
}
```