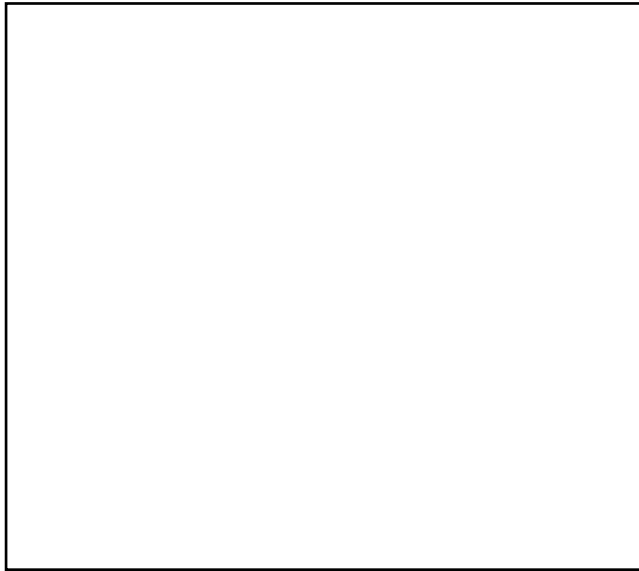


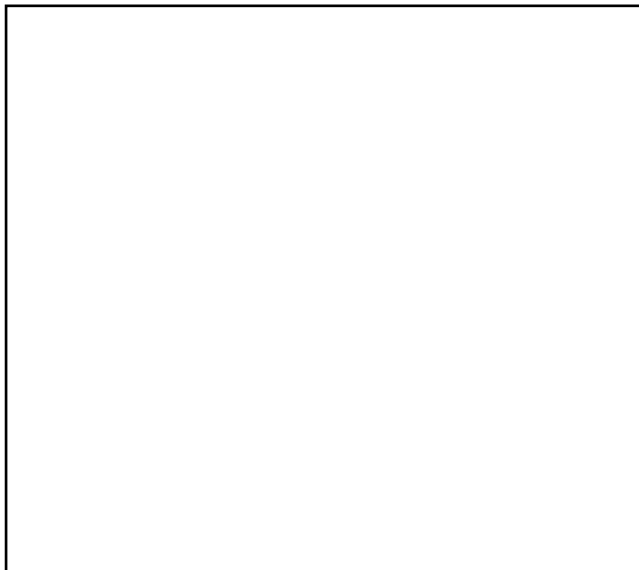
Evidence for Implementation and Testing Unit.

Jaime Lopez
Cohort E20

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



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



I.T 3 - Example of searching

```
27
28 def self.find(id)
29   sql = "SELECT * FROM customers
30   WHERE id = s1;"
31   values = [id]
32   customer_array = SqlRunner.run(sql, values)
33   return Customer.new(customer_array.first) unless customer_array.first == nil
34 end
```

start_point — DetectiveAzul@Reikumi-Zero — ..k/start_point — -zsh —

```
+ start_point git:(master) * ruby console.rb
"I'm looking for an entry for an specific ID on the database"
"Then, I the function creates an object that stores the found data"
#<Customer:0x007f93b23d9f08 @name="Pawel", @funds=45, @id=29>
+ start_point git:(master) *
```

I.T 4 - Example of sorting

```
untitled    customer.rb    codeclan_0/name.sql    console.rb
23 def self.all_by_name()
24   sql = "SELECT * FROM customers
25   ORDER BY name;"
26   customers = SqlRunner.run(sql)
27   return customers.map { |customer| Customer.new(customer) }
28 end
```

models/customer.rb 29:1 (6, 184) LF UTF-8 Ruby master 3 files

start_point — DetectiveAzul@Reikumi-Zero — ..k/start_point — -zsh — 162x19

```
+ start_point git:(master) * ruby console.rb
"Normal output of the names after running the method to find all the customers"
Pawel Sian Bethany Anthony
"Output of the names after using the method to find customers ordered by name"
Anthony, Bethany, Pawel, Sian, S
+ start_point git:(master) *
```

I.T	I.T 6	Demonstrate the use of a hash in a program. Take screenshots of:
		<ul style="list-style-type: none">• A hash in a program• A function that uses the hash• The result of the function running

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

The screenshot shows a Ruby IDE with two files: `room.rb` and `start_point.rb`. The `room.rb` file defines a `Room` class with attributes `number`, `size`, `entry_fee`, `occupants`, `playlist`, and `till`. The `start_point.rb` file defines a function `add_guest_to_occupants` that takes a `guest` and a `room` object as arguments. The function checks if there is enough space in the room and if the guest can afford the entry fee. If both conditions are met, the guest is added to the `occupants` array, and the function returns `true`. Otherwise, it returns `false`.

```

# room.rb
class Room
  attr_reader :number, :size, :entry_fee
  attr_accessor :occupants, :playlist, :till
  def initialize(number, size, till, playlist)
    @number = number
    @size = size
    @playlist = playlist
    @occupants = []
    @entry_fee = 11
    @till = till
  end
end

# start_point.rb
def add_guest_to_occupants(guest)
  if enough_space?() && guest.can_entry_fee?(entry_fee)
    @occupants << guest
    add_money_to_till(@entry_fee)
    return true
  end
  return false
end

```

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

The screenshot shows a Ruby IDE with two files: `pet_shop.rb` and `start_point/pet_shop.rb`. The `pet_shop.rb` file defines a hash `pet_shop` containing information about five pets: `Sir Percy`, `King Dogemagus`, `Sir Lancelot`, `Arthur`, and `Arthor`. Each pet has attributes `name`, `pet_type`, `breed`, and `price`. The `start_point/pet_shop.rb` file defines a function `add_pet_to_shop` that takes a `pet` object and a `pet_shop` hash as arguments. The function adds the pet to the `pet_shop` hash and returns the updated hash.

```

# pet_shop.rb
pet_shop = {
  pet1: {
    name: "Sir Percy",
    pet_type: :cat,
    breed: "British Shorthair",
    price: 300
  },
  pet2: {
    name: "King Dogemagus",
    pet_type: :cat,
    breed: "British Shorthair",
    price: 300
  },
  pet3: {
    name: "Sir Lancelot",
    pet_type: :dog,
    breed: "Bosky",
    price: 1000
  },
  pet4: {
    name: "Arthur",
    pet_type: :dog,
    breed: "Husky",
    price: 300
  },
  pet5: {
    name: "Arthor",
    pet_type: :dog,
    breed: "Husky",
    price: 300
  }
}

# start_point/pet_shop.rb
def add_pet_to_shop(pet, pet_shop)
  pet_shop[pet.name] = pet
  return pet_shop
end

```

I.T 7 - Example of polymorphism in a program