

Evidence for Implementation and Testing Unit.

Jo Malo, E21

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

```
package film_memorabilia.filmTitleParents;

import ...

public class JawsItem extends Item {
    private String film;
    private ItemType type;
    private Edition edition;

    private ArrayList<StarWarsItem> starWarsItems;

    public JawsItem(String film, ItemType type, Edition edition, String itemDescription, int buyPrice, int shipping, int marketValue, String purchaseDate) {
        super(type, edition, itemDescription, buyPrice, shipping, marketValue, purchaseDate);
        this.film = film;
        this.type = type;
        this.edition = edition;

        starWarsItems = new ArrayList<StarWarsItem>();
    }

    public String getFilm() { return this.film; }

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

    public Edition getEdition() { return this.edition; }
}
```

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

```
package film_memorabilia.filmTitleParents;

import ...

public class MuppetsItem extends Item {
    private ItemType type;
    private Edition edition;

    public MuppetsItem(ItemType type, Edition edition, String itemDescription, int buyPrice, int shipping, int marketValue, String purchaseDate) {
        super(type, edition, itemDescription, buyPrice, shipping, marketValue, purchaseDate);
        this.type = type;
        this.edition = edition;
    }

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

    public Edition getEdition() { return this.edition; }
}
```

I.T 3 - Example of searching

```
def film()  
  sql = "SELECT * FROM films WHERE id = $1"  
  values = [@film_id]  
  film_hash = SqlRunner.run( sql, values).first()  
  return Film.new(film_hash)  
end
```

```
+ cinema_homework git:(master) x ruby db/console.rb  
  
From: /Users/jomalo/codeclan_work/week_03/day_5/cinema_homework/db/console.rb @ line 29 :  
  
24: ticket2.save()  
25:  
26:  
27:  
28: binding.pry  
=> 29: nil  
  
[1] pry(main)> ticket1.customer()  
=> #<Customer:0x007fc0518c7350 @funds="100", @id=55, @name="Barry Hope">
```

I.T 4 – Example of sorting I.T 5 - Example of an array, a function that uses an array and the result

```
def self.order()  
  sql = 'SELECT * FROM customers  
        ORDER BY name'  
  customer_hashes = SqlRunner.run(sql)  
  return Customer.map_items(customer_hashes)  
end
```

```
[1] pry(main)> Customer.all()  
=> [#<Customer:0x007fc706253eb8 @funds="100", @id=61, @name="Barry Hope">,  
    #<Customer:0x007fc706253dc8 @funds="150", @id=62, @name="Jude Gaff">,  
    #<Customer:0x007fc706253c60 @funds="200", @id=63, @name="Gerry Hill">]  
[2] pry(main)> Customer.order()  
=> [#<Customer:0x007fc706222840 @funds="100", @id=61, @name="Barry Hope">,  
    #<Customer:0x007fc7062225c0 @funds="200", @id=63, @name="Gerry Hill">,  
    #<Customer:0x007fc706222278 @funds="150", @id=62, @name="Jude Gaff">]
```

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

```
def sum_array(numbers)
  total = 0
  for number in numbers
    total += number
  end
  return total
end

def test_sum_array
  result = sum_array( [ 1,2,3,4,5 ] )
  assert_equal( 15, result )
end
```

specs — jomalo@js-MacBook-Pro — zsh — 123x28

..t_point/specs/java_project ..21_classnotes -zsh ...

```
def first
  4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
  → specs clear
  if
    → specs ruby my_functions_spec.rb
    Run options: --seed 40340
  end
  # Running:
end
return...
end
Finished in 0.001073s, 3727.8659 runs/s, 3727.8659 assertions/s.

def get
  4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
  → specs
```

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

```
user.rb location.rb console.rb quest_advisor.s... sql_runner.rb visit.rb
```

```
23 visit1 = Visit.new({ 'user_id' => user1.id, 'location_id' =>
  * location1.id, 'review' => '0 stars, far too hot'})
24 visit1.save()
25 visit2 = Visit.new({ 'user_id' => user3.id, 'location_id' =>
  * location1.id, 'review' => '5 stars, would visit again if I could'})
26 visit2.save()
27 visit3 = Visit.new({ 'user_id' => user1.id, 'location_id' =>
  * location2.id, 'review' => '4 stars, plenty of beer available'})
28 visit3.save()
29 visit4 = Visit.new({ 'user_id' => user2.id, 'location_id' =>
  * location2.id, 'review' => '3 stars, too crowded, could not find my
  * wizard friend'})
30 visit4.save()
31
32
33
34
```

```
47 #class methods
48 def self.all()
49   sql = "SELECT * FROM visits"
50   visit_hashes = SqlRunner.run(sql)
51   visits = visit_hashes.map { |visit| Visit.new( visit ) }
52   return visits
53 end
54
55 # def self.delete_all()
56 #   sql = "DELETE FROM visits"
57 #   SqlRunner.run(sql)
58 # end
59
60 end
```

ruby db — ruby console.rb — ruby console.rb — 166x29

ruby/java_project ..21_classnotes -zsh ...

```
From: /Users/jomalo/codeclan_work/week_03/day_4/quest_advisor_start/db/console.rb @ line 33 :

28: visit3.save()
29: visit4 = Visit.new({ 'user_id' => user2.id, 'location_id' => location2.id, 'review' => '3 stars, too crowded, could not find my wizard friend'})
30: visit4.save()
31:
32: binding.pry
=> 33: nil

[1] pry(main)> Visit.all
=> [#<Visit:0x007ff49ca14968 @id=1, @location_id=1, @review="0 stars, far too hot", @user_id=1>,
#<Visit:0x007ff49ca146e0 @id=2, @location_id=1, @review="5 stars, would visit again if I could", @user_id=3>,
#<Visit:0x007ff49ca145a8 @id=3, @location_id=2, @review="4 stars, plenty of beer available", @user_id=1>,
#<Visit:0x007ff49ca143c8 @id=4, @location_id=2, @review="3 stars, too crowded, could not find my wizard friend", @user_id=2>,
#<Visit:0x007ff49ca141e8 @id=5, @location_id=3, @review="0 stars, far too hot", @user_id=4>,
#<Visit:0x007ff49ca144e0 @id=6, @location_id=3, @review="5 stars, would visit again if I could", @user_id=6>,
#<Visit:0x007ff49ca144e0 @id=7, @location_id=4, @review="4 stars, plenty of beer available", @user_id=4>,
#<Visit:0x007ff49ca0f888 @id=8, @location_id=4, @review="3 stars, too crowded, could not find my wizard friend", @user_id=5>,
#<Visit:0x007ff49ca0e038 @id=9, @location_id=5, @review="0 stars, far too hot", @user_id=7>,
#<Visit:0x007ff49ca0e658 @id=10, @location_id=5, @review="5 stars, would visit again if I could", @user_id=9>,
#<Visit:0x007ff49ca0d238 @id=11, @location_id=6, @review="4 stars, plenty of beer available", @user_id=7>,
#<Visit:0x007ff49ca0ce98 @id=12, @location_id=6, @review="3 stars, too crowded, could not find my wizard friend", @user_id=8>,
#<Visit:0x007ff49ca06c00 @id=13, @location_id=7, @review="0 stars, far too hot", @user_id=10>,
#<Visit:0x007ff49ca06660 @id=14, @location_id=7, @review="5 stars, would visit again if I could", @user_id=12>,
#<Visit:0x007ff49ca05da0 @id=15, @location_id=8, @review="4 stars, plenty of beer available", @user_id=10>,
#<Visit:0x007ff49ca9ff680 @id=16, @location_id=8, @review="3 stars, too crowded, could not find my wizard friend", @user_id=11>]
[2] pry(main)>
```

I.T 7 - Example of polymorphism in a program



The screenshot shows an IDE with several tabs open: test.java, Person.java, Customer.java, Dodgems.java, ThemePark.java, DodgemsTest.java, IPayment.java, and CustomerTest.java. The Person.java tab is active, displaying the following Java code:

```
public abstract class Person {  
    private int age;  
    public int wallet;  
    public double height;  
  
    public Person(int age, int wallet, double height){  
        this.age = age;  
        this.wallet = wallet;  
        this.height = height;  
    }  
  
    public int getAge(){  
        return this.age;  
    }  
  
    public int getWallet(){  
        return this.wallet;  
    }  
  
    public double getHeight(){  
        return this.height;  
    }  
}
```