

## I.T 1

-encapsulation in a program

```
public abstract class Meal implements Serializable{

    private String food;
    private int calories;

    public Meal(String food, int calories){
        this.food = food;
        this.calories = calories;
    }

    public String getFood(){ return this.food; }

    public int getCalories(){ return this.calories; }

    public void setFood(String food) { this.food = food; }

    public void setCalories(int calories) { this.calories = calories; }
}
```

## I.T 2

-A Class

```
public abstract class Instrument implements IPlayable, ISellable{

    private Type type;
    private String model;
    private String colour;
    private String material;
    private double buyPrice;
    private double sellPrice;

    public Instrument(Type type, String model, String colour, String material, double buyPrice, double sellPrice){
        this.type = type;
        this.model = model;
        this.colour = colour;
        this.material = material;
        this.buyPrice = buyPrice;
        this.sellPrice = sellPrice;
    }

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

    public String getModel() {
        return this.model;
    }

    public String getColour() {
        return this.colour;
    }
}
```

-A Class that inherits from the previous class

```
public class Guitar extends Instrument{

    private int strings;

    public Guitar(Type type, String model, String colour, String material, double buyPrice, double sellPrice, int strings) {
        super(type, model, colour, material, buyPrice, sellPrice);
        this.strings = strings;
    }

    public int getStrings() {
        return this.strings;
    }

    public String play(){
        return "Strumming away";
    }
}
```

```

public class GuitarTest {

    Guitar guitar;

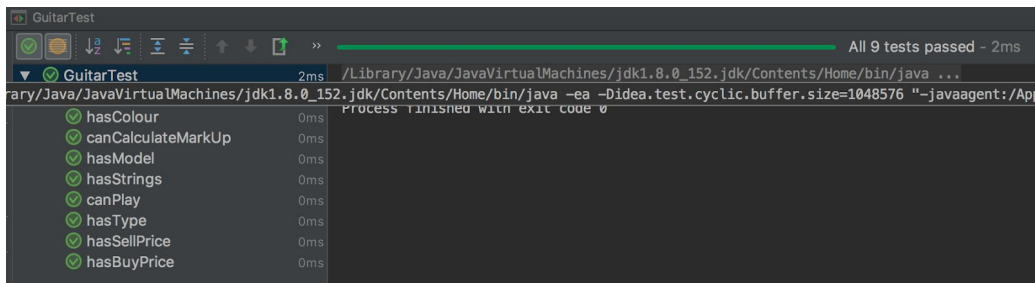
    @Before
    public void before(){
        guitar = new Guitar(Type.STRING, model: "Yamaha GL1", colour: "Brown", material: "Wood", buyPrice: 44, sellPrice: 62, strings: 6);
    }

    @Test
    public void hasType(){
        assertEquals(Type.STRING, guitar.getType());
    }

    @Test
    public void hasModel(){
        assertEquals( expected: "Yamaha GL1", guitar.getModel());
    }

    @Test
    public void hasColour(){
        assertEquals( expected: "Brown", guitar.getColour());
    }
}

```



### I. T 3

Function that searches data:

```

def Transaction.find(id)
  sql = "SELECT * FROM transactions
        WHERE id = $1;"
  values = [id]
  transaction_hash = SqlRunner.run(sql, values).first
  return Transaction.new(transaction_hash)
end

```

The result of the function running:

```

[1] pry(main)> Transaction.find(1)
=> #<Transaction:0x007ff83c25b200
  @amount=10.0,
  @category_id=1,
  @comment="Lost so went to MacDonalds!",
  @id=1,
  @transaction_date="2017-12-19",
  @vendor_id=1>

```

## I. T 4

Function that sorts data:

```
ComputerObjectView.prototype.sortByDate = function () {  
  this.computerObjects.sort(function (earliest, latest) {  
    return earliest.date - latest.date;  
  });  
}
```

```
computerObjectView.sortByDate();  
console.log(computerObjectView.computerObjects);
```

The result of the function running:

```
▼ Array(26)  
  0: {_id: "5aafbeb202a1d71655f42897", id: "co60127", date: 1775, name: "Calculating machi  
  1: {_id: "5aafbeb202a1d71655f4289c", id: "co62748", date: 1847, name: "Babbage's Differ  
  2: {_id: "5aafbeb202a1d71655f42899", id: "co60390", date: 1870, name: "Thomas De Colmar'  
  3: {_id: "5aafbeb202a1d71655f42898", id: "co60113", date: 1892, name: "Brunsviga calcula  
  4: {_id: "5aafbeb202a1d71655f42894", id: "co8408693", date: 1935, name: "Mechanical anal  
  5: {_id: "5aafbeb202a1d71655f4289a", id: "co64128", date: 1939, name: "Four-rotor Enigma  
  6: {_id: "5aafbeb202a1d71655f4289f", id: "co62427", date: 1940, name: "Components from C  
  7: {_id: "5aafbeb202a1d71655f42895", id: "co62556", date: 1940, name: "Aircraft bomb-ais  
  8: {_id: "5aafbeb202a1d71655f4289b", id: "co62349", date: 1949, name: "Automatic Computi  
  9: {_id: "5aafbeb202a1d71655f42893", id: "co8015289", date: 1958, name: "sub unit from l  
 10: {_id: "5aafbeb202a1d71655f4289e", id: "co8359400", date: 1963, name: "computer mouse  
 11: {_id: "5aafbeb202a1d71655f42892", id: "co8061113", date: 1965, name: "DEC PDP-8 mini  
 12: {_id: "5aafbeb202a1d71655f42890", id: "co63204", date: 1972, name: "A sectioned Sinc  
 13: {_id: "5aafbeb202a1d71655f428a0", id: "co503422", date: 1976, name: "Personal Comput  
 14: {_id: "5aafbeb202a1d71655f4288f", id: "co8094235", date: 1983, name: "Nintendo Enter  
 15: {_id: "5aafbeb202a1d71655f4289d", id: "co8401352", date: 1985, name: "Windows 1.0 op  
 16: {_id: "5aafbeb202a1d71655f428a1", id: "co8035886", date: 1989, name: "Intel 486 micr  
 17: {_id: "5aafbeb202a1d71655f4288e", id: "co8194710", date: 1990, name: "Sega Master S  
 18: {_id: "5aafbeb202a1d71655f4288b", id: "co8361832", date: 1992, name: "Nokia 1011 mot  
 19: {_id: "5aafbeb202a1d71655f4288d", id: "co8362946", date: 1994, name: "'Simon' mobile  
 20: {_id: "5aafbeb202a1d71655f428a2", id: "co8430789", date: 1997, name: "IBM ThinkPad i  
 21: {_id: "5aafbeb202a1d71655f428a4", id: "co8184137", date: 1998, name: "Nintendo Game  
 22: {_id: "5aafbeb202a1d71655f428a3", id: "co8361071", date: 2008, name: "Apple 'iPhone  
 23: {_id: "5aafbeb202a1d71655f4288c", id: "co8361046", date: 2012, name: "SIM card for C  
 24: {_id: "5aafbeb202a1d71655f42896", id: "co8361038", date: 2014, name: "Fiberoptic int  
 25: {_id: "5aafbeb202a1d71655f42891", id: "co8401258", date: 2014, name: "Sony 'SmartBar  
    length: 26  
    __proto__: Array(0)
```

## I. T 5

An array in a program:

```
@customers = [  
  {  
    name: "Craig",  
    pets: [],  
    cash: 1000  
  },  
  {  
    name: "Zsolt",  
    pets: [],  
    cash: 50  
  }  
]
```

A function that uses the array:

```
def customer_pet_count(customer)  
  return customer[:pets].count  
end
```

The result of the function running:

```
def test_customer_pet_count
  count = customer_pet_count(@customers[0])
  assert_equal(0, count)
  p count
end
```

→ week\_01\_pet\_shop git:(master) × ruby specs/pet\_shop\_spec.rb  
Run options: --seed 54931

# Running:

.....0  
....

Finished in 0.001510s, 13245.0344 runs/s, 17218.5447 assertions/s.]

20 runs, 26 assertions, 0 failures, 0 errors, 0 skips

## I. T 6

A hash in a program:

```
@pet_shop = {
  pets: [
    {
      name: "Sir Percy",
      pet_type: :cat,
      breed: "British Shorthair",
      price: 500
    },
    {
      name: "King Bagdemagus",
      pet_type: :cat,
      breed: "British Shorthair",
      price: 500
    },
    {
      name: "Sir Lancelot",
      pet_type: :dog,
      breed: "Pomsky",
      price: 1000,
    },
    {
      name: "Arthur",
      pet_type: :dog,
      breed: "Husky",
      price: 900,
    },
    {
      name: "Tristan",
      pet_type: :dog,
      breed: "Basset Hound",
      price: 800,
    },
    {
      name: "Merlin",
      pet_type: :cat,
      breed: "Egyptian Mau",
      price: 1500,
    }
  ],
  admin: {
    total_cash: 1000,
    pets_sold: 0,
  },
  name: "Camelot of Pets"
}
```

A function that uses the hash:

```
def pet_shop_name(pet_shop)
  return pet_shop[:name]
end
```

The result of the function running:

```
def test_pet_shop_name
  name = pet_shop_name(@pet_shop)
  assert_equal("Camelot of Pets", name)
  p name
end
```

```
[-> week_01_pet_shop git:(master) ✕ ruby specs/pet_shop_spec.rb ]
Run options: --seed 38346

# Running:

....."Camelot of Pets"
.....

Finished in 0.001464s, 13661.2024 runs/s, 17759.5632 assertions/s.

20 runs, 26 assertions, 0 failures, 0 errors, 0 skips
[-> week_01_pet_shop git:(master) ✕ ]
```

## I.T 7

-use of polymorphism in a program

```
public class MusicShop {

    private String name;
    private ArrayList<ISellable> stock;
    private double budget;

    public MusicShop(String name, double budget){
        this.name = name;
        this.stock = new ArrayList<>();
        this.budget = budget;
    }

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

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

    public void addToStock(ISellable item) {
        if(item.getBuyPrice() <= getBudget()) {
            this.stock.add(item);
            this.budget -= item.getBuyPrice();
        }
    }
}
```

```

public abstract class Instrument implements IPlayable, ISellable{

    private Type type;
    private String model;
    private String colour;
    private String material;
    private double buyPrice;
    private double sellPrice;

    public Instrument(Type type, String model, String colour, String material, double buyPrice, double sellPrice){
        this.type = type;
        this.model = model;
        this.colour = colour;
        this.material = material;
        this.buyPrice = buyPrice;
        this.sellPrice = sellPrice;
    }

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

    public String getModel() { return this.model; }

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

    public String getMaterial() {
        return this.material;
    }

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

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

    public double calculateMarkUp(){
        return this.sellPrice - this.buyPrice;
    }
}

```

```

public class Saxophone extends Instrument {

    String size;

    public Saxophone(Type type, String model, String colour, String material, double buyPrice, double sellPrice, String size) {
        super(type, model, colour, material, buyPrice, sellPrice);
        this.size = size;
    }

    public String getSize() { return this.size; }

    public String play() { return "Puffing away"; }
}

```

```

public class Tshirt implements ISellable{

    private String size;
    private String description;
    private double buyPrice;
    private double sellPrice;

    public Tshirt(String size, String description, double buyPrice, double sellPrice){
        this.size = size;
        this.description = description;
        this.buyPrice = buyPrice;
        this.sellPrice = sellPrice;
    }

    public String getSize() { return this.size; }

    public String getDescription() { return this.description; }

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

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

    public double calculateMarkUp() { return this.sellPrice - this.buyPrice; }
}

```

```
public interface ISellable {  
    double getBuyPrice();  
    double getSellPrice();  
    double calculateMarkUp();  
}
```