

Evidence for Implementation and Testing Unit.

Yonatan Satat

E16

25/11/2017

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

```
public abstract class Animal {  
    private int cashValue;  
  
    public Animal(int cashValue) { this.cashValue = cashValue; }  
  
    public int getCashValue() {  
        return cashValue;  
    }  
}
```

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

```
public abstract class Animal {  
    private int cashValue;  
  
    public Animal(int cashValue) { this.cashValue = cashValue; }  
  
    public int getCashValue() {  
        return cashValue;  
    }  
}
```

```
public class Lion extends Animal {  
    public Lion(int cashValue) {  
        super(cashValue);  
    }  
}
```

```
public class LionTest {  
    Lion lion;  
  
    @Before  
    public void before() { lion = new Lion( cashValue: 1000); }  
  
    @Test  
    public void testGetCashValue() {  
        assertEquals( expected: 1000, lion.getCashValue());  
    }  
}
```

1 test passed - 2ms
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...
Process finished with exit code 0

I.T 3 - Example of searching

```

persons = ["Yoni", "Danna", "Yossi", "David", "Callum"]
def search(array)
  result = []
  for person in array
    if person.include?("i")
      result.push(person)
    end
  end
  return result
end
print search(persons)

```

```

➔ PDA ruby search.rb
["Yoni", "Yossi", "David"]%
➔ PDA █

```

I.T 4 - Example of sorting

```

3   fruits = ["banana", "orange", "mango", "strawberry", "apple"]
4
5   def sort_fruits(array)
6     array.sort
7   end
8
9   print sort_fruits(fruits)

```

```

➔ PDA ruby sorting.rb
["apple", "banana", "mango", "orange", "strawberry"]%
➔ PDA █

```

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

```
1  fruits = ["banana", "orange", "mango", "strawberry", "apple"]
2
3  def count_fruits(array)
4    number_of_fruits = 0;
5
6    for fruit in array
7      number_of_fruits += 1
8    end
9
10   return number_of_fruits.to_s + " fruits in the array"
11 end
12
13 puts count_fruits(fruits)
14
```

```
→ PDA ruby array.rb
5 fruits in the array
→ PDA
```

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

```
1  pet1 = {age: 3, name: "fig", sound: "mewoo", cat: true}
2
3  def type_of_pet(petHash)
4    if petHash[:sound] == "mewoo"
5      return "You are a cat, " + petHash[:name] + "!"
6    else
7      return "I don't know who you are"
8    end
9  end
10
11 puts type_of_pet(pet1)
12
```



```
→ PDA ruby hash.rb
You are a cat, fig!
→ PDA
```

I.T 7 - Example of polymorphism in a program.

```
interface Drinkable {  
}
```

```
public class Beer implements Drinkable {  
}
```

```
public class Juice implements Drinkable {  
}
```

```
public class Human {  
    private ArrayList<Drinkable> belly = new ArrayList();  
  
    public void drink(Drinkable drink) {  
        belly.add(drink);  
    }  
  
    public int bellySize() {  
        return belly.size();  
    }  
}
```