

Evidence of Implementation and Testing Unit

Paul McPhail Stevenson

Cohort E21

I.T 5 Demonstrate the use of an array in a program. Take screenshots of:

- An array in a program - *@occupancy is an empty array in the left panel.*
- A function that uses the array - *The central panel shows three functions.*
- The result of the function running - *The right hand panel shows the tests working on the specs file.*

```
1 require 'pry'
2 class Rooms
3   attr_reader :name, :playlist, :occupancy,
4   :title, :price
5
6   def initialize(name, playlist, occupancy,
7   title, price)
8     @name = name
9     @playlist = playlist
10    @occupancy = []
11    @title = title
12    @price = price
13  end
14
15  def add_guest(guest)
16    @occupancy << guest
17  end
18
19  def check_wallet(value)
20    return @wallet >= value
21  end
22
23  def pay_entrance_fee(fee)
24    @wallet -= fee
25  end
26
27  # def add_money_to_till(price)
28  #   @till += price
29  # end
30
31 end
32
33 def test_room_has_occupancy
34   assert_equal(@occupancy, @room.occupancy)
35 end
36
37 def test_number_of_guests
38   assert_equal(0, @room.occupancy().count)
39 end
40
41 def test_add_guest_to_occupancy
42   @room.add_guest(guest1)
43   assert_equal(1, @room.occupancy().count)
44 end
45
46 def test_guest_has_a_wallet
47   assert_equal(15, @guest1.wallet)
48 end
49
50 def test_check_wallet_true
51   assert_equal(true,
52     @guest1.check_wallet(5))
53 end
54
55 def test_check_wallet_false
56   assert_equal(false,
57     @guest1.check_wallet(30))
58 end
59
60 def test_pay_entrance_fee
61   @guest1.pay_entrance_fee(5)
62   assert_equal(10, @guest1.wallet)
63 end
64
65 ##CANNOT SET UP PROPERLY
66 # def test_add_money_to_till
67 #   @rooms.money_to_till(5)
68 #   assert_equal(105, @room.till)
69 # end
70
71 end
```

```
Last login: Sat Apr 28 11:44:14 on tty801
- cd codeclan_work/Week_02/d2
- d2 ..
- Week_02 ls
d1 d3 homework
d2 d4 weekend_homework
- Week_02 cd weekend_homework
- weekend_homework git:(master) atom .
- weekend_homework git:(master) atom .
- weekend_homework git:(master) cd specs
- specs git:(master) x ruby rooms_specs.rb
Run options: --seed 16689
# Running:
.....
Finished in 0.002222s, 5400.5401 runs/s, 540
0.5401 assertions/s.
12 runs, 12 assertions, 0 failures, 0 errors
0 skips
- specs git:(master) x []
```

I.T 6 Demonstrate the use of a hash in a program. Take screenshots of:

- A hash in a program - *The left side shows a hash of UK countries.*
- A function that uses the hash - *Northern Ireland is inserted into the hash.*
- The result of the function running - *The right panel is the printed results.*

```
1 united_kingdom = [
2
3   {
4     name: "Scotland",
5     population: 5295000,
6     capital: "Edinburgh"
7   },
8   {
9     name: "Wales",
10    population: 3063000,
11    capital: "Swansea"
12  },
13  {
14    name: "England",
15    population: 53610000,
16    capital: "London"
17  }
18 ]
19
20 united_kingdom[1][:capital] = rep
21 lace("Cardiff")
22
23 united_kingdom << {name:
24   "Northern Ireland",
25   population: 181000,
26   capital: "Belfast"}
27
28 for name in united_kingdom
29   p name[name]
30 end
31
32 total = 0
33 for population in united_kingdom
34   total +=
35     population[:population]
36 end
37
38 p total
```

```
Last login: Sat Apr 28 11:44:14 on tty801
- cd codeclan_work/Week_02/d2
- d2 ..
- Week_02 ls
d1 d3 homework
d2 d4 weekend_homework
- Week_02 cd weekend_homework
- weekend_homework git:(master) atom .
- weekend_homework git:(master) atom .
- weekend_homework git:(master) cd specs
- specs git:(master) x ruby rooms_specs.rb
Run options: --seed 19689
# Running:
.....
Finished in 0.002222s, 5400.5401 runs/s, 5400.5401 assertions/s.
12 runs, 12 assertions, 0 failures, 0 errors, 0 skips
- specs git:(master) x ..
- weekend_homework git:(master) x ..
- Week_02 cd d3
- d3 ls
pub_lab
- d3 atom .
- d3 ../././
- codeclan_work cd Week_01/Homework/Homework d3
- Homework d3 git:(master) x ruby task3.rb
"Scotland"
"Wales"
"England"
"Northern Ireland"
63179680
- Homework d3 git:(master) x ..
```

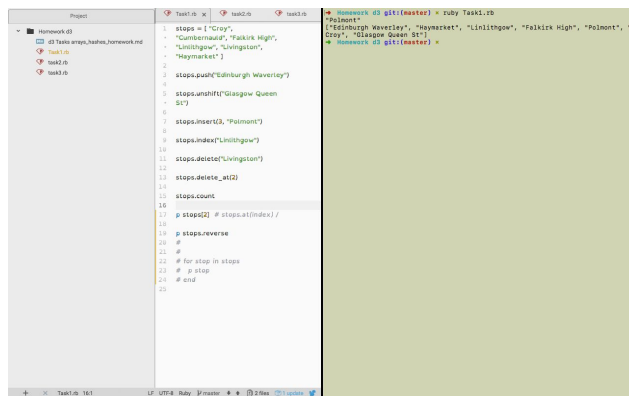
I.T 3 Demonstrate searching data in a program. Take screenshots of:

- Function that searches data
- The result of the function running

I.T 4 Demonstrate sorting data in a program. Take screenshots of:

- Function that sorts data
- The result of the function running

The panel shows data being searched by the index method, and reversed. Results displayed in the terminal window.



```
1 stops = ["Croy",
2           "Sunderland", "Falkirk High",
3           "Linthgow", "Livingston",
4           "Haymarket"]
5 stops.push("Edinburgh Waverley")
6 stops.unshift("Glasgow Queen
7           St")
8 stops.insert(3, "Palmont")
9 stops.index("Linthgow")
10 stops.delete("Livingston")
11 stops.delete_at(2)
12 stops.count
13
14 p stops[2] # stops.at(index) /
15
16 p stops.reverse
17 #
18 # for stop in stops
19 #   p stop
20 # end
```

The screenshot shows a Ruby IDE with a project named 'Task1.3'. The file 'task1.3.rb' is open, displaying the code above. The code defines an array of bus stops, performs several operations (push, unshift, insert, index, delete, delete_at, count), and then prints the element at index 2 and the reversed array. The IDE interface includes a file explorer on the left, a code editor in the center, and a terminal window at the bottom.