



You can view this report online at : <https://www.hackerrank.com/x/tests/545669/candidates/11420173/report>

Full Name: alvin zablan  
Email: azablan@appacademy.io  
Test Name: Ruby Assessment 2 Practice  
Taken On: 19 Oct 2019 10:04:48 PDT  
Time Taken: 7 min 48 sec/ 80 min  
Invited by: Ronil Bhatia  
Tags Score:

100%  
54/54

scored in **Ruby Assessment 2 Practice** in 7 min 48 sec on 19 Oct 2019 10:04:48 PDT

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Vehicle Inheritance - Vehicle methods > Multiple Choice	33 sec	3/ 3	✓
Q2	Vehicle Inheritance - Car methods > Multiple Choice	23 sec	3/ 3	✓
Q3	Vehicle Inheritance - Train Methods > Multiple Choice	19 sec	3/ 3	✓
Q4	Vehicle Inheritance - Sedan Methods > Multiple Choice	14 sec	3/ 3	✓
Q5	Vehicle Inheritance - Prediction > Multiple Choice	9 sec	3/ 3	✓
Q6	Big O - Analysis 1 > Multiple Choice	6 sec	3/ 3	✓
Q7	Big O - Analysis 2 > Multiple Choice	6 sec	3/ 3	✓
Q8	Big O - Analysis 3 > Multiple Choice	8 sec	3/ 3	✓
Q9	Big O - Analysis 4 > Multiple Choice	7 sec	3/ 3	✓
Q10	CSS > Front-end	3 min 53 sec	15/ 15	✓
Q11	Short Answer - Private vs Protected Methods > Subjective	34 sec	6/ 6	⊖
Q12	Short Answer - Set vs Array Data Structure > Subjective	1 min 5 sec	6/ 6	⊖

QUESTION 1  
✓  
Correct Answer

Score 3

Vehicle Inheritance - Vehicle methods > Multiple Choice

QUESTION DESCRIPTION

Which methods does an instance of the `Vehicle` class have access to? (Check all that apply)

Refer to the following code skeleton:

```
class Vehicle
  attr_accessor :speed

  def initialize(speed)
    @speed = speed
  end

  def accelerate(amount)
    self.speed += amount
  end

  def stop
    self.speed = 0
  end
end

class Car < Vehicle
  attr_accessor :location

  def initialize(location, speed)
    super(speed)
    @location = location
  end

  def travel
    self.location += speed
  end
end

class Train < Vehicle
  attr_accessor :route, :station

  def initialize(route, station, speed)
    super(speed)
    @route = route
    @station = station
  end

  def travel
    self.station = station.next
  end

  def switch_route(new_route)
    self.route = new_route
  end
end

class SportsCar < Car
  def travel
    self.location += speed
    puts "VROOM!"
  end
end

class Sedan < Car
  attr_reader :trunk

  def initialize(location, speed)
    super(location, speed)
    @trunk = []
  end

  def load(item)
    trunk.push(item)
  end
end
```

### CANDIDATE ANSWER

**Options:** (Expected answer indicated with a tick)

- ☒ ☐ speed
- ☐ location
- ☐ trunk
- ☐ route
- ☐ station
- ☒ ☐ accelerate
- ☒ ☐ stop
- ☐ travel
- ☐ load
- ☐ switch\_route
- ☒ ☐ speed=

No Comments

### QUESTION 2



Correct Answer

Score 3

### Vehicle Inheritance - Car methods > Multiple Choice

#### QUESTION DESCRIPTION

Which methods does an instance of the `Car` class have access to? (Check all that apply)

Refer to the following code skeleton (note that this code is identical to the previous question):

```
class Vehicle
  attr_accessor :speed

  def initialize(speed)
    @speed = speed
  end

  def accelerate(amount)
    self.speed += amount
  end

  def stop
    self.speed = 0
  end
end

class Car < Vehicle
  attr_accessor :location

  def initialize(location, speed)
    super(speed)
    @location = location
  end

  def travel
    self.location += speed
  end
end

class Train < Vehicle
  attr_accessor :route, :station
```

```

def initialize(route, station, speed)
  super(speed)
  @route = route
  @station = station
end

def travel
  self.station = station.next
end

def switch_route(new_route)
  self.route = new_route
end
end

class SportsCar < Car
  def travel
    self.location += speed
    puts "VROOM!"
  end
end

class Sedan < Car
  attr_reader :trunk

  def initialize(location, speed)
    super(location, speed)
    @trunk = []
  end

  def load(item)
    trunk.push(item)
  end
end

```

#### CANDIDATE ANSWER

**Options:** (Expected answer indicated with a tick)

- ☒ ☒ speed
- ☒ ☒ location
- ☐ trunk
- ☐ route
- ☐ station
- ☒ ☒ accelerate
- ☒ ☒ stop
- ☒ ☒ travel
- ☐ load
- ☒ ☒ speed=

No Comments

#### QUESTION 3



Correct Answer

#### Vehicle Inheritance - Train Methods > Multiple Choice

##### QUESTION DESCRIPTION

Which methods does an instance of the 'Train' class have access to? (Check all that apply)

Refer to the following code skeleton (note that this code is identical to the previous question):

```
class Vehicle
  attr_accessor :speed

  def initialize(speed)
    @speed = speed
  end

  def accelerate(amount)
    self.speed += amount
  end

  def stop
    self.speed = 0
  end
end

class Car < Vehicle
  attr_accessor :location

  def initialize(location, speed)
    super(speed)
    @location = location
  end

  def travel
    self.location += speed
  end
end

class Train < Vehicle
  attr_accessor :route, :station

  def initialize(route, station, speed)
    super(speed)
    @route = route
    @station = station
  end

  def travel
    self.station = station.next
  end

  def switch_route(new_route)
    self.route = new_route
  end
end

class SportsCar < Car
  def travel
    self.location += speed
    puts "VROOM!"
  end
end

class Sedan < Car
  attr_reader :trunk

  def initialize(location, speed)
    super(location, speed)
    @trunk = []
  end

  def load(item)
    trunk.push(item)
  end
end
```

### CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☒ ☒ speed
- ☐ location
- ☐ trunk
- ☒ ☒ route
- ☒ ☒ station
- ☒ ☒ accelerate
- ☒ ☒ stop
- ☒ ☒ travel
- ☐ load
- ☒ ☒ switch\_route

No Comments

#### QUESTION 4



Correct Answer

Score 3

### Vehicle Inheritance - Sedan Methods > Multiple Choice

#### QUESTION DESCRIPTION

Which methods does an instance of the `Sedan` class have access to? (Check all that apply)

Refer to the following code skeleton (note that this code is identical to the previous question):

```
class Vehicle
  attr_accessor :speed

  def initialize(speed)
    @speed = speed
  end

  def accelerate(amount)
    self.speed += amount
  end

  def stop
    self.speed = 0
  end
end

class Car < Vehicle
  attr_accessor :location

  def initialize(location, speed)
    super(speed)
    @location = location
  end

  def travel
    self.location += speed
  end
end

class Train < Vehicle
  attr_accessor :route, :station
```

```

    def initialize(route, station, speed)
      super(speed)
      @route = route
      @station = station
    end

    def travel
      self.station = station.next
    end

    def switch_route(new_route)
      self.route = new_route
    end
  end

  class SportsCar < Car
    def travel
      self.location += speed
      puts "VROOM!"
    end
  end

  class Sedan < Car
    attr_reader :trunk

    def initialize(location, speed)
      super(location, speed)
      @trunk = []
    end

    def load(item)
      trunk.push(item)
    end
  end
end

```

#### CANDIDATE ANSWER

**Options:** (Expected answer indicated with a tick)

- ☒ ☐ speed
- ☒ ☐ location
- ☒ ☐ trunk
- ☐ route
- ☐ station
- ☒ ☐ accelerate
- ☒ ☐ stop
- ☒ ☐ travel
- ☒ ☐ load
- ☐ switch\_route

No Comments

#### QUESTION 5



Correct Answer

Score 3

#### Vehicle Inheritance - Prediction > Multiple Choice

##### QUESTION DESCRIPTION

Consider the following code skeleton (note that this code is identical to the previous question):

```

class Vehicle
  attr_accessor :speed

  def initialize(speed)
    @speed = speed
  end

  def accelerate(amount)
    self.speed += amount
  end

  def stop
    self.speed = 0
  end
end

class Car < Vehicle
  attr_accessor :location

  def initialize(location, speed)
    super(speed)
    @location = location
  end

  def travel
    self.location += speed
  end
end

class Train < Vehicle
  attr_accessor :route, :station

  def initialize(route, station, speed)
    super(speed)
    @route = route
    @station = station
  end

  def travel
    self.station = station.next
  end

  def switch_route(new_route)
    self.route = new_route
  end
end

class SportsCar < Car
  def travel
    self.location += speed
    puts "VROOM!"
  end
end

class Sedan < Car
  attr_reader :trunk

  def initialize(location, speed)
    super(location, speed)
    @trunk = []
  end

  def load(item)
    trunk.push(item)
  end
end

```

What would the following code output?



```
sparty = SportsCar.new(0, 80)
sparty.travel
```

#### CANDIDATE ANSWER

**Options:** (Expected answer indicated with a tick)

- ☐ No output
- ☒ "VROOM!"
- ☐ 80
- ☐ 0

No Comments

#### QUESTION 6



Correct Answer

Score 3

#### Big O - Analysis 1 > Multiple Choice

##### QUESTION DESCRIPTION

Which time complexity best describes the code below?

```
def foo(array)
  slice = array[0..array.length / 3]
  slice.each do |el|
    puts el
  end
end
```

Assume that n is the length of the array.

#### CANDIDATE ANSWER

**Options:** (Expected answer indicated with a tick)

- ☒ ☐ O(n)
- ☐ O(1)
- ☐ O(n^2)
- ☐ O(n^3)

No Comments

**QUESTION 7**

Correct Answer

Score 3

**Big O - Analysis 2** > Multiple Choice**QUESTION DESCRIPTION**

Which time complexity best describes the code below?

```
def foo(n)
  while n >= 1
    puts n
    n = n / 2
  end
end
```

**CANDIDATE ANSWER**

**Options:** (Expected answer indicated with a tick)

- ☐ O(n)
- ☐ O(n^2)
- ☒ O(log(n))
- ☐ O(1)

No Comments

**QUESTION 8**

Correct Answer

Score 3

**Big O - Analysis 3** > Multiple Choice**QUESTION DESCRIPTION**

Which time complexity best describes the code below?

```
def foo(array)
  triplets = []
  array.each do |e1|
    array.each do |e2|
      array.each do |e3|
        triplets << [ e1, e2, e3 ]
      end
    end
  end
end
```

Assume n is the length of the array

**CANDIDATE ANSWER**

**Options:** (Expected answer indicated with a tick)

- ☐  $O(n^2)$
- ☐  $O(n)$
- ☒  $O(n^3)$
- ☐  $O(3^n)$

No Comments

**QUESTION 9**

Correct Answer

Score 3

**Big O - Analysis 4** > Multiple Choice**QUESTION DESCRIPTION**

Which time complexity best describes the code below?

```
def foo(n)
  2.times do
    n.times do
      puts "hello"
    end
  end
end
```

**CANDIDATE ANSWER**

**Options:** (Expected answer indicated with a tick)

- ☐ O(1)
- ☐ O(log(n))
- ☒ O(n)
- ☐ O(2^n)

No Comments

**QUESTION 10**

Correct Answer

Score 15

**CSS** > Front-end**QUESTION DESCRIPTION**

An html skeleton is provided. Apply the following styles:

1. Give a font-size of 15px to all li elements
2. Give a blue background to the element with an id of "list-header"
3. Give red text to all elements with the class of "ruby"
4. Give a yellow background to **only** the span elements that are inside li elements
5. Give the image a height of 50px

Three points awarded are awarded for satisfying each instruction, for a maximum of 15 points.

**You may not edit the HTML skeleton.**

**You can render a preview of your applied styles by clicking the "Render" button in the bottom right.**

**INTERNAL NOTES**

## Scoring


3 points awarded for satisfying each instruction, for a maximum of 15 points

**CANDIDATE ANSWER**

Please open the report on HackerRank for Work to view the candidate's submission

<https://www.hackerrank.com/x/tests/545669/candidates/11420173/report>

No Comments

<div>QUESTION 11</div> <div>Self Evaluation</div>	<div>Short Answer - Private vs Protected Methods &gt; Subjective</div>
<div>Score 6</div>	<div>QUESTION DESCRIPTION</div> <div>What is a difference between 'private' and 'protected' methods?</div> <div>Your response must be one or more complete sentences.</div> <div>INTERNAL NOTES</div> <div><div>Scoring</div><div>Full Credit: response fully identifies a difference</div><div>Half Credit: response only partially identifies a difference</div></div> <div>Possible full credit responses:</div> <div><ul style="list-style-type: none"><li>Private methods cannot be called with an explicit receiver while protected methods may be called with an explicit receiver.</li></ul></div> <div>Half credit examples:</div> <div><ul style="list-style-type: none"><li>A difference between private and protected methods are their receivers.</li></ul></div> <div>CANDIDATE ANSWER</div> <div>Private methods cannot be called with an explicit receiver while protected methods may be called with an explicit receiver.</div> <div>No Comments</div>

**QUESTION 12**

Self Evaluation

Score 6

**Short Answer - Set vs Array Data Structure** > Subjective**QUESTION DESCRIPTION**

Which data structure would be the more preferable to keep track of visited nodes when implementing BFS on a graph: a Set or an Array? Explain your answer.

**Your response must be one or more complete sentences.**

**INTERNAL NOTES****Scoring**

**Full Credit:** response identifies answer with valid explanation

**Half Credit:** response identifies answer without valid explanation

Answer: Set is preferable

Explanation: Set has constant time  $O(1)$  lookup, compared to Arrays linear  $O(n)$  lookup.

**CANDIDATE ANSWER**

A set is preferable because a set has constant time  $O(1)$  lookup, compared to Arrays linear  $O(n)$  lookup.

No Comments