QUIZ 2: RUBY

This quiz covers content from the first two weeks of lectures, slides and readings. The quiz is designed to take less than an hour, and should be taken without the aid of a Ruby interpreter.

Please note that you are allowed only one attempt per question.

ADDITIONAL SUPPORT

Need help preparing? Join a **Mentive live section** with a mentor to answer your questions. Available in various time zones and languages.

Quiz 2-1: Method Calls

(1 point possible)
In the Ruby expression `a.foo`, which calls method `foo` on receiver `a`, which of the following conditions affects whether the call succeeds?

- [x] `a` responds to the `foo` method.
- [x] `a`'s class, or one of its superclasses or included modules, implements `method_missing` in a way that handles the `foo` method.
- [ ] `a` is an instance of a class, rather than the class itself.
- [ ] `a` has the correct type.

?

You have used 0 of 1 submissions

Quiz 2-2: The Send Method

(1 point possible)
What is the result of executing the following code?

```
x = [1,2,3]
x.send :[]=,0,2
x[0] + x.[](1) + x.send(:[],2)
```

- 5
- 6
- 7
- a syntax error or other runtime error

?

You have used 0 of 1 submissions

Quiz 2-3: Method Logistics

(1 point possible)
When the Ruby expression `foo + bar` is evaluated, which statement is ALWAYS true?

- `bar` is passed as an argument to `foo`'s `+` method.
- `foo` is passed as an argument to `bar`'s `+` method.
- An error will occur because `+` is only defined for strings and numbers.
- `foo` and `bar` will be converted to numbers (if necessary) and summed.

?

You have used 0 of 1 submissions

Quiz 2-4: Regular Expressions 1

(1 point possible)

You are given the following short list of movies exported from an Excel comma-separated values (CSV) file. Each entry is a single string that contains the movie name in double quotes, zero or more spaces, and the movie rating in double quotes. For example, here is a list with three entries:

```
    movies = [%q{"Aladdin",   "G"},
              %q{"I, Robot", "PG-13"},
              %q{"Star Wars","PG"}]
```

Your job is to create a regular expression to help parse this list:

```
    movies.each do |movie|
      movie.match(regexp)
      title,rating = $1,$2
    end
    # => for first entry, title should be Aladdin, rating
should be G, WITHOUT the double
    # => quotes
```

You may assume movie titles and ratings never contain double-quote marks. Within a single entry, a variable number of spaces (including 0) may appear between the comma after the title and the opening quote of the rating.

Which of the following regular expressions will accomplish this? Check all that apply.

- [x] `regexp = /"([^"]+)",\s*"([^"]+)"/`
- [x] `regexp = /"(.*)",\s*"(.*)"/`
- [ ] `regexp = /"(.*)", "(.*)"/`
- [ ] `regexp = /(.*),\s*(.*)/`

?

You have used 0 of 1 submissions

Quiz 2-5: Inheritance

(1 point possible)
Class `A` is a subclass of Class `B`. Class `B` is a subclass of class `C`. `a` is an object of class `A`. `b` is an object of class `B`. Which ONE of the following Ruby expressions is NOT true?

- `b.respond_to?('class')`
- `a.superclass == b.class`
- `A.superclass == B`
- `a.class.ancestors.include?(C)`

?

You have used 0 of 1 submissions

Quiz 2-6: Regular Expressions 2

(1 point possible)
A valid DNA sequence consists of any number of A, G, C, and T bases. A valid RNA sequence consists of any number of A, G, C, and U bases. (Valid DNA sequences never contain U, and valid RNA sequences never contain T.) Which regular expression below matches an entire sequence that is either valid DNA or valid RNA?

- `/(^[ACGU]+$|^[AGCT]+$)/`
- `/^[ACGUT]+$/`
- `/^[(ACGU)|(ACGT)]+$/`

- `/^([ACG]|[TU])+$/`

?

You have used 0 of 1 submissions

Quiz 2-7: Classes

(1 point possible)
Which of the following statements about classes in Ruby are true?

- [x] `Array` is an instance of `Class`.
- [ ] When `self` is used within the definition of an instance method, it refers to the current instance of the class.
- [ ] Ruby supports multiple inheritance.
- [ ] Public methods of a class cannot be redefined after an instance of that class is instantiated.

?

You have used 0 of 1 submissions

Quiz 2-8: Attribute Accessors

(1 point possible)
Given the following code:

```
class Book
  attr_accessor :author
  attr_reader :title
  attr_writer :comments
  def initialize(author, title)
    @author = author
    @title = title
    @comments = []
  end
end

book = Book.new("Chuck Palahniuk", "Fight Club")
```

Which of the following snippets of code are valid?

- [x] `"#{book.title} was written by #{book.author}."`
- [ ] `book.comments << "#{book.title} was a good book"`
- [ ] `book.comments.each { |comment| puts comment }`
- [ ] `book.title = "Cooking Club"`

?

You have used 0 of 1 submissions

Quiz 2-9: Agile Lifecycle

(1 point possible)
True or False: The Agile lifecycle is the best model for all software development?

- ○ True
- ◉ False

?

You have used 0 of 1 submissions

Quiz 2-10: Modules

(1 point possible)
Which of these statements are true regarding Modules in Ruby?

1. Modules hold a collection of methods and constants
2. Modules can inherit from classes
3. Modules allow you to share functionality across classes

- (1) and (3)
- (2) and (3)
- (1) and (2)
- (1), (2) and (3)

?

You have used 0 of 1 submissions

Quiz 2-11: String Methods

(1 point possible)
Given a string `s = "Hello: I`m a l33t programmer!!"` what single line of Ruby will produce the array
`["Hello", "m", "a", "l", "t", "programmer"]`

- `s.split(/[^a-zA-Z]/).reject{|e| e == "I" || e.empty?}`
- `s.split(/[^a-zA-Z]/).reject{|e| e == "I"}`
- `s.scan(/[a-zA-Z]/).reject{|e| e == "I" || e.empty?}`
- `s.split(/[^a-z^A-Z]/).reject{|e| e == "I"}`

?

You have used 0 of 1 submissions

Quiz 2-12: Iterators

(1 point possible)
Consider the following code:

```
def mystery_sequence(start1, start2, limit=4)
  yield start1
  yield start2
  nextval = start1 + start2
  1.upto(limit) do
    yield nextval
    nextval, start2 = nextval + start2, nextval
  end
end
```

Assuming `s` is initially an empty array `[]`, which call will cause `s` to have the value: `[2,3,5,8,13,21]` ?

- (•) `mystery_sequence(2,3) do |elt| ; s<<elt; end`
- ( ) `mystery_sequence(2,3).each do |elt| ; s<<elt; end`
- ( ) `mystery_sequence(2,3).do |elt| ; s<<elt; end`
- ( ) `mystery_sequence.each(2,3) do |elt| ; s<<elt; end`

?

You have used 0 of 1 submissions