

CMSC 330 Fall 2016 Quiz #1

Name _____

Discussion Time (circle one) 10am 11am 12noon 1pm 2pm 3pm

Discussion TA (circle one) Alex Austin Ayman Brian Damien Daniel K.
Daniel P. Greg Tammy Tim Vitung Will K.

Instructions

- Do not start this quiz until you are told to do so.
- You have 20 minutes for this quiz.
- This is a closed book quiz. No notes or other aids are allowed.
- For partial credit, show all of your work and clearly indicate your answers.

1. (4 points) Write a Ruby regular expression to match only strings which represent timestamps, which represent a date and an **optional** time. The format of these timestamps is YYYY-MM-DD HH:MM:SS, where the date and time are separated by a single space. Here are some strings which your regex should and should not match:

Should match	Should not match
"2016-09-10 20:26:56"	"abc2016-09-10 20:26:56def"
"2017-04-30"	"2016-09-10 "
"0001-40-77 46:99:00"	"201-609-10 :20:26:56"

Note that you do **not** need to validate the dates and times themselves, just their format.

2. (9 points) What is the output of the following Ruby programs? If there is no output, please write only **NO OUTPUT**. Recall that `foo.inspect` gives the representation of `foo` as it would appear in source code, e.g. `[1,2,3].inspect` is `"[1,2,3]"`.

a) (3 points)

```
arr = []  
arr[3] = "foo"  
puts arr.inspect
```

b) (3 points)

```
arr = [15, 23, 4, 16, 8, 42]  
arr.sort!  
arr.select { |x| x.even? }  
puts arr.inspect
```

c) (3 points)

```
foo = "abcd45efghi" =~ /\d/  
puts $1.inspect
```

3. (7 points) Given the following Ruby `Set` class declaration, implement the indicated methods:

```
class Set
  include Enumerable

  def initialize
    @s = Hash.new(false)
  end

  def insert(val)
    @s[val] = true
  end

  # Your methods are here.

end
```

- a) (3 points) Implement `contains?` for `Set`. It should return `true` if `val` is in the set, and `false` otherwise.

```
def contains?(val)
```

```
end
```

- b) (4 points) Implement the `each` method for `Set`. Recall that `each` takes a code block and yields each member of a collection to it in turn. You may iterate over the elements in any order.

```
def each
```

```
end
```