CS 4700

Written Homework 1  See webpage and Eagle for due dates  (10 points)

Note, these exercises may be done in groups of one, two, or three. Working with someone else is strongly recommended. If more than one person is involved, list all the names on ONE set of answers. Groups may change throughout the term.

Working in groups is a BIG plus for you. Take advantage of it. If you work in groups, you must work in the group for the ENTIRE assignment. It is considered cheating if you work with someone else for some of the answers, but turn in an individual copy of the answers. It is an all or nothing situation. You can't work together on some questions and alone on some. Sometimes I see an individual whose name is listed in two groups. This is strictly forbidden and is considered cheating. You cannot work in two groups. Assignments are due at the beginning of class and should be typed.

Ruby

1. What are advantages and disadvantages of an interpreted language?

2. Ruby claims to be truly object-oriented. In what way is Ruby more object-oriented than other languages that claim to be object-oriented?

3. List three interesting differences between C++ and Ruby.

4. Giving Ruby examples, explain the difference between deep and shallow copy.

5. When code blocks are passed in Ruby, do the variables in the code block have the scope of the caller or the callee? Explain.

6. What is a symbol in Ruby (e.g., :MyName)? Why is a symbol used?

7. How do hashes in Ruby demonstrate the classic time-space tradeoff known to computer scientists?

8. Write the regular expression for each of the following:
   a. Write a regular expression for all strings of 0’s and 1’s in which the string 101 never occurs as a substring. For example, 0000, 11001, and 1100 are legal. 10011011 is illegal.
   b. Write the regular expression for all strings of a's and b's in which the number of b's is divisible by two, such as aabaabbaba
   c. Write the regular expression for all strings of a's and b's in which three b's never occur together.

9. For the following regular expressions, give two examples of what would match and what would not:
   a) /\^[L-Z]\in/ 
   b) /\((4\,[0-3])\ (2\,[0-3])\)/ 
   c) /\([W][L]\)/ 
   d) /<0(x)(D\,[a-f])(A-F)\+/ 

10. We say that the * operator (in Ruby regular expressions) is greedy. Give an example to show what that means.
11. What is the difference between single quotes and double quotes in a Ruby string?
12. Ruby is considered to be strongly typed but not statically typed. Give an example to show what that means.
13. How are arrays implemented in Ruby? Be sure to explain how we are able to add elements within the array or extend the size of the array. (Similarly, deletes are allowed anywhere in the array.)
14. How does one specify a class variable (as opposed to a class instance variable)?
15. What is an array slice and how is it implemented?
16. Write the code to change all occurrences of "good" to "best" in a string.
17. Write the code to create an array from a string such that all the integer groups of the string become elements of the array. For example, the string "I really am 4 good 123 days in 2010" would be used to create the array [4,123,2010]
18. Write the code to examine a string. For the first occurrence of anything of the form MM/DD/YYYY, print out "Month DD Year YYYY" where DD is a two digit integer and YYYY is a four digit integer.
19. Knowing what you know about hash tables, how is it that functions like "each_key" are implemented in Ruby?
20. Explain how "yield" works and give an example where it is useful.

For the following questions, what would be produced? (For this homework, you can obviously run Ruby to find out, but be sure you could answer the question without running Ruby)

21.
```
presidents.pop
presidents.pop
presidents.pop
presidents.unshift("Nixon")
presidents.unshift("Johnson")
presidents.unshift("Kennedy")
presidents.push("Obama")
presidents.each { |i| print i, "\n"}
```

22.
```
a=[34,45,56,2,13,54]
a << [137,89, 5,0]
p = a.length
```

23.
```
a=[34,45,[56, 57],2,13,54]
a.flatten
p a
```

24.
```
presidents = []
presidents[0] = "Washington"
```
presidents[9] = "Harrison"
presidents[15] = "Buchanan"
presidents.each { |i| print i, "\n"}

25.
numbers = ["one", "two", "buckle", "my", "shoe", "three", "four", "shut", "the", "door"]
numbers.each { |i| print i, "\n"}
numbers[2,2]=[]
numbers.each { |i| print i, "\n"}

26.
numbers = ["one", "two", "3", "4", "5", "seven"]
numbers.each { |i| print i, "\n"}
numbers[2,3]=["three", "four", "five", "six"]
numbers.each { |i| print i, "\n"}

**Topic 1**

27. What is true of an interpreter?
   a. An interpreter does the conversion line by line as the program is run
   b. An interpreter is a representation of the system being designed
   c. An interpreter is a general purpose implementation providing very efficient execution
   d. An interpreter converts the code to an intermediate form before executing.

28. Given the following properties of a Class variable in C++, which are static and which are dynamic:
   a. its value
   b. its data type
   c. its name