**Name:anil kolusu Quiz 2**

1. Ruby program when that accepts a date in a month and prints it out the appropriate suffix. For example, for 1 as input, print 1st, 2 as input 2nd, 3 as input 3rdetc
2. <http://docs.ansible.com/ansible/YAMLSyntax.html>

[h](http://ruby-doc.org/stdlib-2.0.0/libdoc/yaml/rdoc/YAML.html" \o ")[http://www.w3schools.com/json/default.asp](http://www.w3schools.com/json/default.asp" \o ")

1. [ttp://ruby-doc.org/stdlib-2.0.0/libdoc/yaml/rdoc/YAML.html](http://ruby-doc.org/stdlib-2.0.0/libdoc/yaml/rdoc/YAML.html" \o ")
2. <https://gist.github.com/timsavery/1657351>
3. <http://ruby-doc.org/stdlib-1.9.3/libdoc/uri/rdoc/URI.html>

2. Write your own ruby program using a case statement

Ans: #!usr/bin/ruby

Puts “enter a number”

age = gets.to\_i

case age

when 0..2

puts “baby”

when 3..6

puts “little child”

when 7..12

puts “child”

when 13..18

puts “youth”

else

puts “adult”

end

output: enter a number

16

youth

3. Ruby Program that iterates numbers with upto loop

Ans: #!usr/bin/ruby

3.upto (7) {|i| puts “hello anil #{i}”} ## I is a variable which starting from 3 upto 7 include.

4. Write a Ruby program to print numbers from 1 to 50 and also in reverse order

Ans: #!usr/bin/ruby

k=1

until k==51

print k, “\n”

k +=1

end

50.downto (1) {|k| puts “#{k}”}

Output:1

2

3

.

.

50

50

49

.

.

1

5. Write your own Ruby program using loops and iterators. Explain the difference between loops, iterators and blocks

Ans:

1. the main difference between loops and iterators is , some times in loops if we are fail to ensure termination condition facing infinity loops it must be terminate only the when condition is true.

For example in loops :k=1

While k=10 ##insted of while k<10

Print k, “\n”

K +=1

end

output:it prints 10 repetedly.

But in iterators it is impossible.we can print upto 10 like this

1.upto (10) {|i| puts #{i}”}

1. #loops

i=0

While i<3

Print i, “\n”

i +=1

end

but we can write in simple way by using iterators like this

#iterator

3.times do puts “hello”

End

Blocks: A block is always invoked from a function with the same name of that block.

def call\_block

puts “start of method”

yield (“hello”, 99) ## it will call block

puts “end of the method”

end

call\_block { ###first this block is executed then it calls method first.

|str, num|puts “Inside the block” +str+” ” + num.to\_s

}

Output: start of the method

Inside the block hello 99

End of the method.

6. Write a Ruby program that loops through a array and checks if a pattern existsin the array elements

Ans:

#!usr/bin/ruby

items = [“Boots”, “Clock”, “Laptop”]

for i in 0..items.lenght -1

if items[i]==”Clock”

puts” the pattern is found”

else

puts “the pattern is not available ”

end

end

out put: the pattern is not available

the pattern is found

the pattern is not available.

7. Write your own Ruby program using a Hash that loops through :

Print all Values while looping with Keys

items = Hash.new()

items [“milk”] =5

items [“eggs”] =10

items [“bread”] =15

puts items [“milk”]

puts items .fetch(“eggs”)

puts items .values\_at “bread”

output:5

10

15

Or puts items.values

Print all Keys while looping through Values

Ans: puts items.keys

Or for only one value

Puts items.rassoc(“5”)

It displays associated key.

Print Keys, Values as pair.

Ans:from the above program we can disply the keys and values as apair like

Puts items.assoc(“eggs”)

Output: eggs

10

8. Write a Ruby program that takes number as input and recursively calculates the power of 2 until the calculated number is less than 10000 and prints the maximum power for that number.

Ans: #!usr/bin/ruby

num = 1

while num < 1000 do

num = num \*2

end

num = num/2

puts “#{num} is the highest power of 2 less then 10000.”

9. Ruby program to convert Celsius temperature to Fahrenheit

Ans:

#!usr/bin/ruby

puts “enter the Celsius value:”

Celsius = gets.to\_i

Puts “ converted to Celsius to fahrenheit=”+(( Celsius+32)\*5/9).to\_s

gets

10. Write a program to create a Calculator class with add(), substract(), multiply() and divide(), then take two numbers and choice of operation from user and display output using objects

Ans: class Calculator

def initialize(a,b)

@a = a

@b = b

end

def add

puts @a + @b

end

def sub

puts @a - @b

end

def mul

puts @a \* @b

end

end

puts "enter 1'st number"

@a=gets.to\_i

puts "enter 2'nd number"

@b=gets.to\_i

cal=Calculator.new(@a,@b)

puts "enter an operator"

ope=gets.chomp

if ope == "+"

cal.add

elsif ope == "-"

cal.sub

elsif ope == "\*"

cal.mul

else

puts "you enter wrong operator"

end

gets

output: enter 1'st number

9

enter 2’nd number

16

Enter an operator

-

7