GEDİZ UNIVERSITY

Computer Engineering Department

COM401
Software Engineering Laboratory

October 21, 2014

LAB-2:

Ruby – Part II

Time: 2 lab hours

Objectives:

Learn

- Blocks
- Idea of modules
- Logic of Regular Expressions
- Inheritance
- Access restriction

Learn how to

- use blocks
- use method_missing
- write some patterns by using Regexes.

Lab Outcomes:

Practice with RubyMine IDE and Ruby and some OOP concepts in Ruby.

Exercises :

1- Write a Ruby script that calculates and writes out the square of integers from 1 to 10 by using "each" block.

2- Write a Ruby script that illustrates the function of "yield".

3- Write a Ruby script that detects the first number greater than 10 in a given set of integers.

4- Write a Ruby script that selects the odd numbers in a given set of integers.

5- Write a Ruby script that converts each element of a string array into uppercase, then return the result back with a new array (use *collect*).

6- Create a class named "MathTest", then define summation, substraction and multiplication methods for it. However, try to all a method named "div" without define it. Then, try to handle with this situation by using *method_missing*.

7- Write a Ruby script that shows a simple regex example in Ruby bay using =~ and *match* method from String class.

8- Write a Regex that matches with first column of numbers in a String.

9- Write a Regex that matches any sequence of exactly two numerical digits at the end of each line.

10- Write a Ruby script substitutes the word "Java" to "Ruby" by using *gsub* and *gsub!*
Sample string can be "Welcome to Java"

11- Write a Ruby script that illustrates accessing captures via regexes.

12- Write a Ruby script that illustrates accessing all matches by using *scan* method.

13- Create a Person class including two attributes : name and surname, then add a to string method to it.

14- Create an Employee class derived from “Person” class. Also, Employee class has name, surname and title/position information. Then, give a grant access to read name and surname, and grant access to write title attribute.

15- Create a Vehicle class, then add a class variable named as "no_of_vehicles" that can keep track of the number of objects created that inherit from Vehicle. Next, create a method to print out the value of this class variable as well.

16- Write a Ruby class named as ‘Person’ that illustrates access modifiers like protected, private.

17- Write a simple Animal, Bird, Fish, and Snake classed that illustrate method overloading.

18- Use the Animal and Snake classes to illustrate private methods in inheritance.