

Lab session 4 – Ruby (part 1)

| | |
|-------------------|---|
| Unit | Programming languages: principles and design (6G6Z1110) Programming languages – SE frameworks (6G6Z1115) |
| Lecturer | Rob Frampton |
| Week | 4 |
| Portfolio element | This lab is not part of the portfolio. |

Description

The aim of this lab exercise is to practise the several Ruby elements as studied in the lecture “Ruby – part 1”. By the end of this lab session, you should:

1. have learnt how to write and run small pieces of code in Ruby, and
2. have practised several basic Ruby elements.

Everything you need will be in the lectures (apart from Exercise 5b), however links to relevant parts of Ruby documentation are provided which you may find useful.

Exercises

Exercise 1a:

The program below prints the contents of the file *lyrics.txt* to the screen:

```
File.open("lyrics.txt", "r") do |infile|
  while line = infile.gets
    puts "#{line}"
  end
end
```

Modify the program so that it prints a line number before each line, such that the output given the file *lyrics.txt* (on Moodle) would be:

```
1: Never gonna give you up,
2: Never gonna let you down
3: Never gonna run around and desert you
4: Never gonna make you cry,
5: Never gonna say goodbye
6: Never gonna tell a lie and hurt you
```

Exercise 1b:

Again, modify the program such that each line is reversed as shown below:

```
1: ,pu uoy evig annog reven
2: nwod uoy tel annog reven
3: uoy tresed dna dnuora nur annog reven
4: ,yrc uoy ekam annog reven
5: eybdoog yas annog reven
6: uoy truh dna eil a llet annog reven
```

Hint: if you find that you need to remove the newline at the end of each line, use [chomp](#).

Exercise 1c:

Again, modify the program so that the order of the words in each line is reversed instead. You will probably need the [split](#) method of `String` and the [join](#) and [reverse](#) methods of `Array`

```
1: up, you give gonna Never
2: down you let gonna Never
3: you desert and around run gonna Never
4: cry, you make gonna Never
5: goodbye say gonna Never
6: you hurt and lie a tell gonna Never
```

Exercise 2:

Write a new program that reads in *lyrics.txt* and then displays the number of times each word occurs. You will need to use a [Hash](#). The output should be (note the order of words is not important):

```
Never    occurred 6 times
gonna    occurred 6 times
give     occurred 1 times
you      occurred 5 times
up       occurred 1 times
let      occurred 1 times
down     occurred 1 times
run      occurred 1 times
around   occurred 1 times
and       occurred 2 times
desert   occurred 1 times
make     occurred 1 times
cry      occurred 1 times
say      occurred 1 times
goodbye  occurred 1 times
tell     occurred 1 times
a        occurred 1 times
lie      occurred 1 times
hurt     occurred 1 times
```

Hint: You can print a tab character ("\t ") after the word to help line up the output

Exercise 3:

The code below prints all the even numbers between 0 and 100.

```
puts (0..100).select { |a| a % 2 == 0 }
```

Modify the code so it prints all the *odd* numbers between 0 and 50.

Exercise 4:

Complete the program below so that it only prints words longer than `minLength` from the file `words.txt` (available on Moodle):

```
minLength = 7
words = []
File.open( "words.txt", "r") do |infile|
  while (word = infile.gets)
    words << word
  end
end
words = words.select <insert code here>
puts "there are #{words.length} words longer than #{minLength} letters"
```

The output should be:

there are 135 words longer than 7 letters

Exercise 5a:

Create a program in Ruby using a “for” statement that outputs the numbers between 1 and 200 but replaces any number divisible by three with the word "**Fizz**", and any number divisible by five with the word "**Buzz**".

Hint: use the modulo operator (%) to test divisibility

Exercise 5b:

Implement Exercise 5a but using the [map](#) method on Range. Your program will look like this:

```
(1..200).map <insert code here>
```

Exercise 6: Extension Exercise

Using Ruby solve problem 17 from Project Euler [<http://ProjectEuler.net>]:

“If the numbers 1 to 5 are written out in words: one, two, three, four, five, then there are 3 + 3 + 5 + 4 + 4 = 19 letters used in total. If all the numbers from 1 to 1000 (one thousand) inclusive were written out in words, how many letters would be used?

NOTE: Do not count spaces or hyphens. For example, 342 (three hundred and forty-two) contains 23 letters and 115 (one hundred and fifteen) contains 20 letters. The use of "and" when writing out numbers is in compliance with British usage.”