# Software Engineering. Learning from errors.

**By: Georgi Ivanov, Momchil Angelov, Dimitar NestOrov**
**Organization: Illuminati**
**Team Name: Happy Satan ☺**
**Date: October 27th, 2014**

One of the main problems was that the people who made their own fixtures to check if their programs would work in real life scenarios were not that many. If they spared 2 or 3 minutes to create fixtures they would've seen what their programs do right and what they do wrong.

Most common mistakes:

Wrong CSV name.
Programs don't check if the name of the file is correct.

# Program 1: 11b class

```
require 'csv'
a = Array.new
h = Hash.new
Dir.glob("#{ARGV[0]}/*.rb") do |dir_file_name_1|
        Dir.glob("#{ARGV[1]}/*.rb") do |dir_file_name_2|

                file_name_1 = dir_file_name_1.split(/\//).last.to_s
                file_name_2 = dir_file_name_2.split(/\//).last.to_s

                if(file_name_1 != file_name_2)
                        file_name = file_name_1
                        digit = file_name.split(/_/).last.split(/\./).first.to_s
                        first_name = file_name.split(/_/).first.to_s
                        full_first_name = first_name + digit
                        full_first_name = full_first_name.to_s
                        tmp = file_name.split("#{first_name}_")
                        full_last_name = tmp.last.split(/_/).first.to_s + digit
                        full_last_name = full_last_name.to_s
                        h[full_last_name] = full_first_name

                end
        end
end

CSV.open("results.csv", "w") do |csv|
        a = h.sort
        a.each do |element|
                csv << element
        end
end
```

I don't even know from where to start. Don't even try compiling it. It's tired and emotional. Like look at it http://imgur.com/Y0WrLap .Its abysmal. Worst/5, because I don't know the scaling. I cannot into comprehend!

# Program 2: 11b class

```
#Develop a program named FirstName_LastName_ClassNumber_1eea4f.rb

#1. you are given an argument for a folder with files;
#1.1 if there are other arguments they should be discarded
#2. file names in this folder are in the form First_Last_digits.rb;
#3. find all the students that have 5 letters in their second name;
#4. Sort the result by Last Name ASC.
#5. Produce a result in CSV format named result.csv:

#               FirstName1,LastName1
#               FirstName2,LastName2
#               ...
#               FirstNameN,LastNameN

        require 'csv'
        students_names = []
        Dir.glob("#{ARGV[0]}/**/*.rb") do |current_file|

        name = current_file.split('/').last.split(/_/)
        if name[1].length == 5
                if not students_names.include?(["#{name[1]}", "#{name[0]}"]) then
                        students_names << (["#{name[1]}", "#{name[0]}"])
                end
        end
        end
        CSV.open("result.csv", "w") do |csv|
                students_names.sort.each do |last, first|
                        csv << ["#{first}",  "#{last}"]
                end
        end
```

With this specimen the situation is kind of better. This specimen tried to first get the file, split it and see if it has 5 symbols in his last name. Which is good if he actually checked whether they are in the format or not. His program needs to be rewritten by actually 1st asking whether its in the format or not, because "if name[1].length == 5" returns an error because u cant get length from a nill class (if the file is Momchil.rb, the name[1] is a nill class). Please help me.

Mark: 3/5 which is on the better spectrum because his code is readable, and his error isn't actually that bad- not having fixtures was his bigger problem.

To fix this, I suggest using this if:

```
if !name[0] ||  !name[1] ||  name[2].to_i == 0
next
else if name[1].length == 5 and so on with his code.
```

This will make his program check first then ask for size, which will fix nillclass error.
His sort is sorting by 1st element too, so we have to fix it too.
In his original idea, the sort and writing in CSV is in one line:

```
students_names.sort.each do |last, first|
                        csv << ["#{first}",  "#{last}"]
```

but we can split it in 2 lines:

```
students_names = students_names.sort_by{|el| el[1]}
```

and

```
students_names.each do |last, first|
                        csv << ["#{first}",  "#{last}"]
```

# Program 3: 11b class

Develop a program named FirstName_LastName_ClassNumber_f8b0d9.rb

1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form First_Last_digits.rb
3. find  the students that are only in the first folder and not in the second. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
4. Sort the result by Last name ;
5. Produce a result in CSV format named result.csv:

```
LastName1,FirstName1
LastName2,FirstName2
...
LastNameN,FirstNameN
```

```
require 'csv'
results = Hash.new
results.compare_by_identity
def is_number(str)
        str[/[0-9]+/] == str
end
Dir.glob("#{ARGV[0]}/*.rb") do |path1|
        filename1 = path1.split(/\//).last
        if filename1.count("_") == 2
                firstname1 = filename1.split("_").first
```

```
                lastname1 = filename1.split("_")[1]
                digit1 = filename1.split("_")[2].split(".").first
                if is_number(digit1)
                        flag = 0
                        Dir.glob("#{ARGV[1]}/*.rb") do |path2|
                                filename2 = path2.split(/\//).last
                                if filename2.count("_") == 2
                                        digit2 = filename2.split("_")[2].split(".").first
                                        if is_number(digit2)
                                                name1 = firstname1 + lastname1
                                name2 = filename2.split("_").first + filename2.split("_")[1]
                                                if name1 == name2
                                                        flag = 1
                                                        break
                                                end
                                        end
                                end
                        end
                        if flag == 0
                                results[lastname1] = firstname1
                        end
                end
        end
end

CSV.open("result.csv", "w") do |csv|
        results.sort_by{|key, val| key}.each do |el|
                csv << el
        end
end
```

Mark: 2/5 because the readability is so low, but it works, which is a bonus.

# Program 4: 11b class

Develop a program named FirstName_LastName_ClassNumber_e0ea9c.rb

1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form First_Last_digits.rb;
3. find the students that are only in the second folder and not in the first. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
4. Sort the result by First name ;

5. Produce a result in CSV format named result.csv:

```
LastName1,FirstName1
LastName2,FirstName2
...
LastNameN,FirstNameN
```

```
require 'csv'
student = Array.new
student1 = Array.new

Dir.glob(ARGV[0]+"/**/*.*").each do |file_name1|
        file_name = file_name1.split("/").last
        first_name = file_name.split("/").last.split("_").first
        p first_name
        last_name = file_name.split("/").last.split("_",2).last.split("_").first
        #task = file_name.split("_").last.split(".").first
        student << ["#{first_name}", "#{last_name}"]
end

Dir.glob(ARGV[1]+"/**/*.*").each do  |file_name1|
        file_name = file_name1.split("/").last
        first_name = file_name.split("/").last.split("_").first
        p first_name
        last_name = file_name.split("/").last.split("_",2).last.split("_").first
        #task = file_name.split("_").last.split(".").first
        student1 << ["#{first_name}", "#{last_name}"]
end

CSV.open("result.csv", "w") do |csv|
        student.each do |fn, ln|
                student1.each do |fn1, ln1|
                        if fn != fn1
                                if ln != ln1
                                        csv << ["#{fn1}", "#{ln1}"]
                                end
                        end
                end
        end
end
```

The CSV is kind of odd. The code can be made better by using Array operators- & and -

Student 2= student1 – student

instead of  the whole writing in the CSV ordeal.

student2.sort_by{|el| el[1]} for the sorting
And her code is ready to go!
I rate it 3/5 because its hard to read but still needs so little to work.

# Program 5: 11b class

```
#Develop a program named FirstName_LastName_ClassNumber_f70059.rb

#1. you are given two arguments for a folders with files;
#1.1 if there are other arguments they should be discarded;
#2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names
excluding extension. If there are duplicates the file must be written only once.;
#3. Calculate the length of their names (including extensions) divided by 2 rounded to the
smalles number;
#4. Sort the result by File name ;
#5. Produce a result in CSV format named result.csv:

#                       File1,3
#                       File2,4
#                       ...
#                       FileN,3

require 'csv'
hash = Hash.new
count = 0
        Dir.glob(ARGV[0] + "/*.rb") do |file|

                first = file.split("/").last
                puts first

                size = first.length
                i = 0
                first.each do |element|

                        c = first[i].chr
                        if element == 0 || element == 1 || element == 2 || element == 3 || element
== 4 || element == 5 || element == 6 || element == 7 || element == 8 || element == 9
                        count +=1

                        end
                end
                puts count
        end

        Dir.glob(ARGV[1] +"/*.rb") do |secFile|
                sec = secFile.split(/\//).last
                #puts sec

        end

        CSV.open("result.csv", "w") do |csv|
                hash.sort_by{|key,val| key}.each do |element|
```

```
            csv << element
            end
        end
```

He forgot to use the split("_") method.
He has far too many mistakes. This work looks half-baked.0/5, not even readability can save him!

# Program 6: 11b class

```
#Develop a program named FirstName_LastName_ClassNumber_627d43.r#
#
#1. you are given two arguments for a folders with files;
#1.1 if there are other arguments they should be discarded;
#2. file names in this folders are in the form First_Last_digits.rb;
#3. find  the students that are only in the first folder and not in the second. A student is in both folders if it there is a file with the same First and Last #Name. Digits might be different;
#4. Sort the result by Last name ;
#5. Produce a result in CSV format named result.csv:
#
#       LastName1,FirstName1
#       LastName2,FirstName2
#       ...
#       LastNameN,FirstNameN



require 'csv'
name_array = Array.new()
name_array2 = Array.new()
support_array = Array.new()
support_array2 = Array.new()
i = 0
dir1 = ARGV[0]
dir2= ARGV[1]

Dir.glob("#{dir1}/*.*") do |file|
        name_array[i] = file.split(/\//).last
        i += 1
end
count = i
i = 0
Dir.glob("#{dir2}/*.*") do |file2|
        name_array2[i] = file2.split(/\//).last
        i += 1
end
i = 0
for check in i..count
        if name_array[check] != name_array2[check]
```

```
                        support_array[i] = name_array[check]
                        support_array2[i] = name_array2[check]
                        i += 1
                        puts support_array
                        puts support_array2
                        CSV.open("result.csv", "w") do |csv|
                                support_array.each do |element|
                                        csv << [element]
                                end
                        end
                        CSV.open("result.csv", "w") do |csv|
                                support_array2.each do |element2|
                                                csv << [element2]
                                end
                        end
        end
end
```

Do we have the same tutor as them? I mean, this is literally C. Please learn array operators.
3/5 for being able to get the general idea. Can't think of a short solution though.
The problem is outputting too much and the actual filter mechanism doesn't work.

# Program 7: 11b class

```
#Develop a program named FirstName_LastName_ClassNumber_d77aee.rb
#
#1. you are given two arguments for a folders with files;
#1.1 if there are other arguments they should be discarded;
#2. Find all the files from both folders that are not in the format
FirsrName_LastName_digit.rb. If there are duplicates the file #must be written only once. If
two files are of the same lenght those files should be sorted in ASC order;
#3. Calculate the length of their names (including extensions).;
#4. Sort the result by length ;
#5. Produce a result in CSV format named result.csv:
#
#                       File1,3
#                       File2,4
#                       ...
#                       FileN,3

require 'csv'

first_folder = ARGV.shift
second_folder = ARGV.shift || "err"
names_hash = Hash.new
```

```
Dir.glob(first_folder+"/*.*").each do |text_file|
	text_file = text_file.split("/").last
	if (text_file.split("_").length == 3) then
		first_name = text_file.split("_")[0]
		second_name = text_file.split("_")[1]
		diggit = text_file.split("_")[2].split(/\./).first
		if (diggit.to_i.to_s != diggit) then names_hash[text_file] = text_file.length end
		if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
		if (second_name =~ /\d/) then names_hash[text_file] = text_file.length end
	else
		names_hash[text_file] = text_file.length
	end
end

if second_folder != "err"
	Dir.glob(second_folder+"/*.*").each do |text_file|
		text_file = text_file.split("/").last
		if (text_file.split("_").length == 3) then
			first_name = text_file.split("_")[0]
			second_name = text_file.split("_")[1]
			diggit = text_file.split("_")[2].split(/\./).first
			if (diggit.to_i.to_s != diggit) then names_hash[text_file] =
text_file.length end
			if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
			if (second_name =~ /\d/) then names_hash[text_file] = text_file.length
end
		else
			names_hash[text_file] = text_file.length
		end
	end
end

names_hash = Hash[names_hash.sort_by{|k,v| k} ]
names_hash = Hash[names_hash.sort_by{|k,v| v} ]

puts names_hash

CSV.open("results.csv","w") do |csv|
	names_hash.each do |element|
		csv << element
	end
end
```

Wrong output. The sorting is wrong although I never get to it. At least he sorted his hash before the CSV writing.
Its hard to fix the code. No immediate solution.
Mark: 3/5

# Program 8: 11b class

Develop a program named FirstName_LastName_ClassNumber_041472.rb

1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form First_Last_digits.rb;
3. find the students that are only in the first folder and not in the second. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
4. Sort the result by Last name ;
5. Produce a result in CSV format named result.csv:

```
LastName1,FirstName1
LastName2,FirstName2
...
LastNameN,FirstNameN
```

```
students_first_dir = Array.new
students_second_dir = Array.new

for i in 0..1

        directory = ARGV[i]
        if ARGV[i].split(//).last(1).to_s == "/"
                directory += "**/*.rb"
        else
                directory += "/**/*.rb"
        end

        Dir.glob(directory).each do |dir|
                student = dir.split(/\//)
                if i == 0
                        students_first_dir.push(student)
                else
                        students_second_dir.push(student)
                end
        end
end

studentcsv = Array.new

students_first_dir.each do |std|
        match = 0
        students_second_dir.each do |std2|
                name = std.last.split(/_/)

                name2 = std2.last.split(/_/)
```

```
                for i in 0..1
                        if name[i] == name2[i]
                                match = 1
                        end
                end

        end
        studentcsv.push(name[1], name[2])
end

CSV.open("result.csv", "w") do |csv|
        studentcsv.each do |string|
                csv << string
        end
end
```

Wrong. Name is not defined. Its error prone!
I could not find a short solution.
Mark: 1/5