RUBY LAB EXERCISE NO: 9

**NAME : KARTHICK VENKATESH T C**

**REG.NO : 18MIS1102**

1. Define a module **STUDENT** with the following details:

Variables : **reg number, name**

Functions : **read()-** to initialize the values

**Display()-** to display the values

Define a another module **FACULTY** with the following details:

Variables : **employee Id, name, school**

Functions : **read1()-** to initialize the values

**Display1()-** to display the values

Define a class **COURSE ALLOCATION** with the details **: course code, Slot** and includes all the details of both the modules **STUDENT** and **FACULTY**.

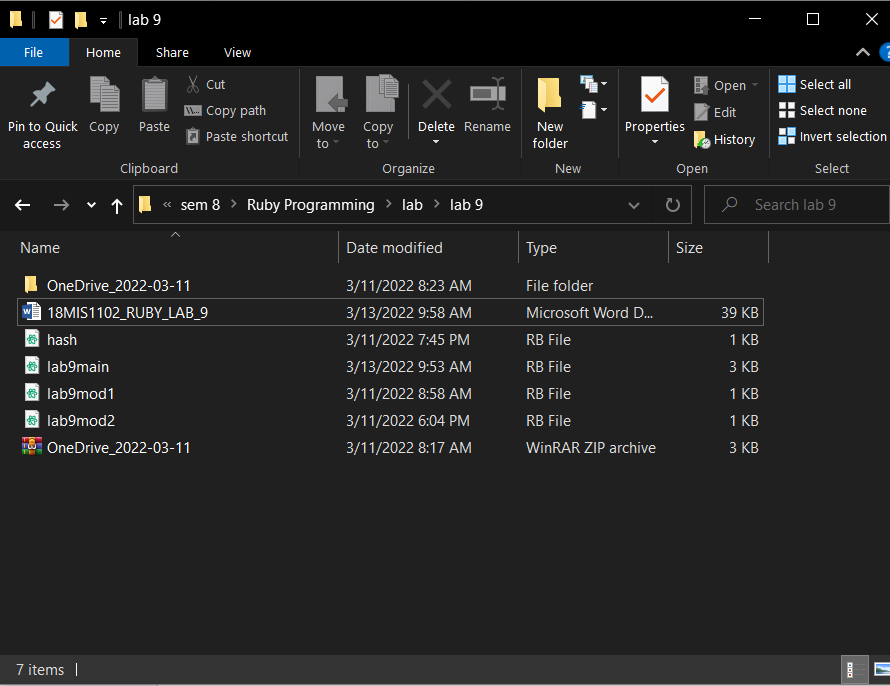
Perform the following operation :

1. Create N objects of COURSE ALLOCATION class. Initialize the values using Constructor .
2. Display all the records
3. Display the faculty names who are handling same subjects . For each subject display all faculty names.
4. How many students are registered for each faculty for a specific course. Accept the course code from the user.
5. List out all course name with the Faculty name in each slot.
6. Display total number fo courses handled by the faculty of each school.

**SOLUTION:**

**CODE:**

**FILE LOCATION OF 2 MODULES:**



**STUDENT MODULE:**

module Student

attr\_accessor :Reg,:Name

def Student.read(r,na)

@Reg = r

@Name = na

end

def Student.display

puts("Reg.no is #{@Reg}")

puts("Student Name is #{@Name}")

end

end

**FACULTY MODULE:**

module Faculty

attr\_accessor :eid,:name,:school

def Faculty.read1(eid,name,school)

@eid = eid

@name = name

@school = school

end

def Faculty.display1()

puts("Faculty emp id: #{@eid}")

puts("Faculty name: #{@name}")

puts("Faculty School: #{@school}")

end

end

**MAIN MODULE CLASS COURSE\_ALLOCATION:**

$LOAD\_PATH << '.'

require"lab9mod1"

require"lab9mod2"

class Course\_allocation

include Student

include Faculty

@@subj = Hash.new

@@han = Hash.new

@@stud = []

def initialize(cc,slot,reg,name,eid,fname,school)

@cc = cc

@slot = slot

@fname = fname

@name = name

Student.read(reg,name)

Faculty.read1(eid,fname,school)

x = []

x.push(@cc,@fname,@name)

@@stud.push(x)

if @@han.empty?

@@han[@fname] = 1

else

if @@han.has\_key?(@fname)

@@han[@fname] = @@han[@fname]+1

else

@@han[@fname] = 1

end

end

if @@subj.empty?

x = []

x.push(@fname)

@@subj[@cc] = x

else

if @@subj.has\_key?(@cc)

#x = []

#x.push(@fname)

#@@subj[@cc] = x

#@@subj[@cc] = (@@subj.values\_at(@cc)).push(@fname)

@@subj[@cc] = @@subj[@cc].push(@fname)

else

x = []

x.push(@fname)

@@subj[@cc] = x

#@@subj[@cc] = @@subj.values\_at(@cc).push(@fname)

end

end

end

def self.spcourse(x)

for i in 0...@@stud.length() do

if @@stud[i][0] == x

puts("Faculty: #{@@stud[i][1]} and student: #{@@stud[i][2]} and course\_code: #{@@stud[i][0]}")

end

end

end

def display

Student.display()

Faculty.display1()

end

def self.disp()

puts(@@subj)

puts("courses handled by faculty: #{@@han}")

end

end

c = Course\_allocation.new(1223,"b2","18mis1102","karthick",57797,"prakash","scope")

c1 = Course\_allocation.new(1223,"b2","18mis1079","kammu",57785,"imran","scope")

c2 = Course\_allocation.new(1023,"b1","18bce1121","karnesh",57785,"imran","scope")

c.display

Course\_allocation.spcourse(1223)

Course\_allocation.disp()

**OUTPUT:**

