**Group Activity 3; CS 3060**

**Names** of students in your group: Nicholas W, Dylan S, Ethan F, Nicholas B, Tyson S

Points: 10 (with 2 bonus points)

Most of the code is given. You need to fill in a little to complete the code.

**Task 1**: (3 points) Consider hash h (where key is employee id and value is employee name) as follows. Sort this hash as required. Submit the test results in a readme.

*a = {"12" => “adam smith”, "13" => “sd roy”, "22" => “ed nelson”}*

*b1 = a.sort\_by{|x| x[1]. }.to\_h # sort h according to the length of employee names*

*p b1*

*b2 = a.sort{|x, y| x[ ] <=>y[ ]} # sort h according to the employee names*

*p b2*

**Task 2**: (3 points) Say function *foo* gets an array (named *a1*) of strings as input. Then, in function foo do the following. (a) Use the *find\_all* or *select* function to filter out strings from *a1*, which start with character “c” (b) Use *inject* function to compute the total length of all strings in *array1*.

*def foo (a1) #* *a1 is an array of* strings

*b = a1.select{|x| x[ ] != “c”} #* strings in a1 which do not start with character “c” go to b

*p b*

*totalLength = a1.inject( ){|s, item| s + item. }*

*p totalLength*

*end*

Test your code by calling *foo* as follows and present the result in a readme.

*foo([“aadfg”, “cdf”, “kjhg”, “ce”])*

**Task 3**: (3 points) Run the following code. What is the outcome? Give a brief explanation. Submit the test results in a readme.

*class Array*

*def find*

*for i in 0...size*

*item = self[i]*

*return item if yield(item)*

*end*

*return nil*

*end*

*end*

*a = [1, 3, 5, 7, 9]*

*b = a.find {|v| v\*2 > 10 }*

*p b*

**Task 4**: (3 points) Read the following definition of Animal, Human and Dog class. Why is Animal an abstract class? We need a precise answer. Furthermore, note that Human and Dog are concrete classes. Now create a random list of 10 animals (i.e. Human or Dog chosen randomly). Traverse the list, and on each animal invoke the *makes\_sound* function. Submit the test results in a readme.

*class Animal # this is an abstract class*

*attr\_accessor :age, :sound*

*def initialize(age)*

*@age = age*

*end*

*def makes\_sound*

*list = []*

*for i in 0..10*

*toss = rand(2)*

*if toss == 0*

*h = Human.new( )*

*list.push(h)*

*else*

*d = Dog.new( )*

*list.push(d)*

*end*

*end*

*sound*

*end*

*end*

*class Human < Animal*

*attr\_accessor :legs*

*def initialize (age)*

*super*

*@legs = 2*

*@sound = "talk"*

*end*

*end*

*class Dog < Animal*

*attr\_accessor :legs*

*def initialize (age)*

*super*

*@legs = 4*

*@sound = "bark"*

*end*

*end*

**Submission**: Submit one copy (per group) of work (two files: completed ruby code in this word document and a readme file with the test results) to Canvas.