# Programming Paradigms Exercise 1 - Ruby

Theodoros Chondrogiannis

2nd Semester 2016/17

1. Implement the following methods:

   (a) A method `arraycopyodd` that gets an array as a parameter and returns the elements found at odd positions in a new, smaller array. For the example, for the array `[5,4,7,8,3]` print `4` and `8`.

   (b) A method `arrayappend` that gets two arrays as a parameter, appends the second array to the first one and returns all the elements in a new array. For example, given the arrays `[5,4]` and `[7,8,3]`, the array `[5,4,7,8,3]` will be returned.

   (c) A method `selectsort` that gets an array as an argument and returns the array sorted in increasing order. Use the *selection sort* algorithm (see `http://en.wikipedia.org/wiki/Selection_sort`).

   (d) A method `arrayunion` that gets two arrays as a parameter and returns a new array which contains the union of the elements contained in the two input arrays and is ordered. For example, given the arrays `[5,4,2,6,9]` and `[7,5,6,6,2]`, the array `[2,4,5,6,7,9]` will be returned.

2. Write a program that opens a file and outputs the file line by line with the line number in front of the line, e.g.

```
1: x = 0
2: while x < y
3: z = compute(x)
4: ...
```

   Hints: `File.new(filename, mode)` creates a `File` object and opens a file named `filename` according to the specified mode. For example, `f = File.new("testfile", "r")` opens a file named `testfile` in read-only mode. The method `gets` fetches a line from the file (returns false if there are no more lines), while the method `close` closes the file.

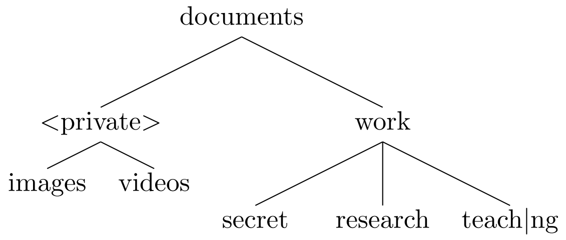
3. In this exercise we will create a class representing directories and extend the functionality of the class by including mixins of modules.

(a) Create a class `Directory` with the attributes `name` (a string) and `subdirectories` (a list of `Directory` objects). Ensure that objects are printed using their name (*hint:* `puts obj` uses `obj.to_s()` to convert the object to a string).

(b) The name of a file or a directory must not include any of the following characters: / >< | : &. Create a Ruby module `FilesystemObject` with the function `isValid?` to check the whether a given directory name is valid or not. Include the module `FilesystemObject` to the mixins of the `Directory` class.

(c) Create a module `TreeNode` with a method `bfsprint()` which prints out all non-root tree nodes using recursive breadth-first-search and a method `getChildren` which returns the node's child nodes. Include the module to the mixins of the `Directory` class. Then add a method to the module `TreeNode` called `bfs` such that `documents.bfs(){|x| puts x}` produces the same output as `documents.bfsprint()`.

(d) Create an object `documents` for the following directory tree:



and output all subdirectories which do not have a valid name.