

## Programming and Problem Solving

Assignment 2 --- Due Sunday, March 24, 2021

### Part I

**Please read carefully:** You must submit the answers to all the questions below. However, this part will not be marked. Nonetheless, failing to submit this part fully will result in you missing 50% of the total mark of the assignment.

#### Question 1

- a) Given an array of integers of any size,  $n \geq 1$ , write an algorithm as a **pseudo code** (not a program!) that would find all the mirrored pairs from the array. A mirrored pair (a, b) produces  $a + b = 0$  i.e., sum of the elements is zero. For instance, given [1, 3, -5, 9, -2, -4, 5, 7, 4, 6], the algorithm will return [-5, 5] and [-4, 4].
- b) Find a simple solution with time complexity of  $O(n^2)$
- c) Can this task be performed in  $O(n \log n)$  or  $O(\log n)$ ? If yes, please provide the pseudo code for your approach. If not, explain why it is not possible.
- d) Is a solution in  $O(n)$  even a possibility? If yes, please provide the pseudo code for your approach. If not, explain why it is not possible.

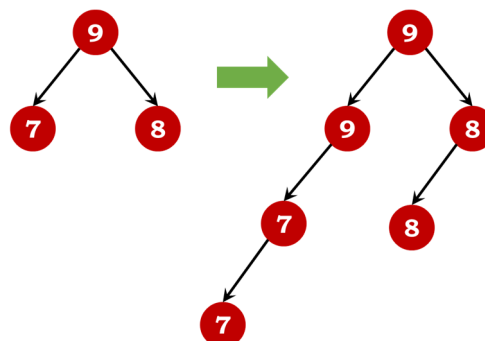
#### Question 2

Given a collection of  $n$  numbers, write an algorithm, **using pseudo code** that will output all possible combinations of  $r$  numbers. For instance, given {1,2,3,4,5}, algorithm must output (1,2), (1,3), (1,4), (1, 5), (2,3), (2,4), (2,5), (3,4), (3,5) and (4,5) for  $r = 2$ .

- a) Write an iterative as well as a recursive solution for this problem.
- b) What is the time complexity of your algorithm, in terms of Big-O?
- c) What changes will you make to the algorithm to output all permutations instead. The output for the same input will now contain (1,2) as well as (2,1).

#### Question 3

Develop a **well-documented pseudo code** that inserts a duplicate node as the left child of the node. A sample tree transformation is provided below.



Is it guaranteed to have the resulting tree as a BST if the input was a BST?

## Part II

**Purpose:** The purpose of this assignment is to allow you practice Exception Handling, and File I/O, as well as other previous object-oriented concepts.

Data is the future and as such we will try and create a data driven application.

## Assignment 2

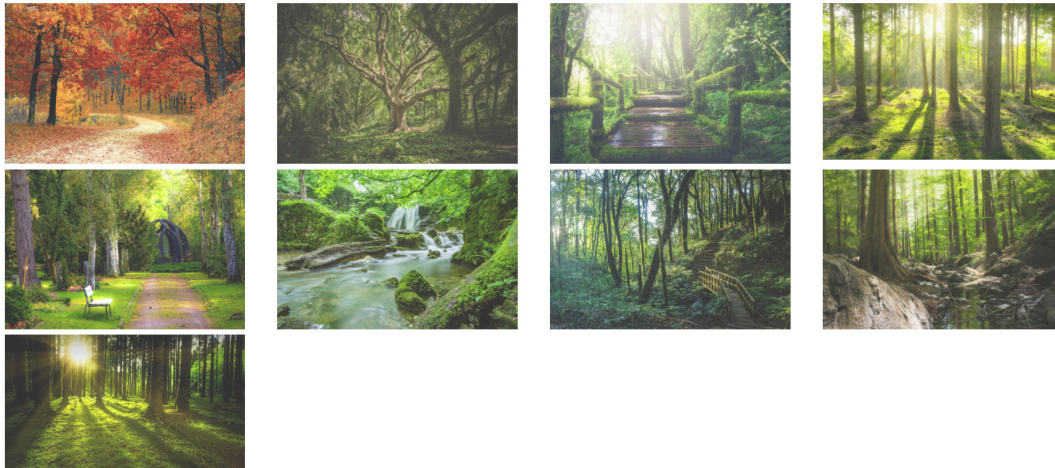


Figure 1. An automatically generated web page from data (Images are from <https://www.pexels.com>)

In this assignment, you will be creating a Java based system for automated web page generation from data provided to you on the fly. Firstly, you will have to list the directory and file structure at a given path and then you will have to process all the files present at that path. A detailed task description is given below:

1. List all the directories, subdirectories and files present at a path. You should assume that there is a folder named **Data** which will be available in the same directory as your .java file and will contain all the files that you must process. You should open a file named **"log.txt"** and write the detailed file structure present in the Data folder in it. You must take the **recursive** approach to list all files and directories present in the Data folder. For example, a sample log of directory system is shown in figure 2. You must use **exception handling** while using the File class.

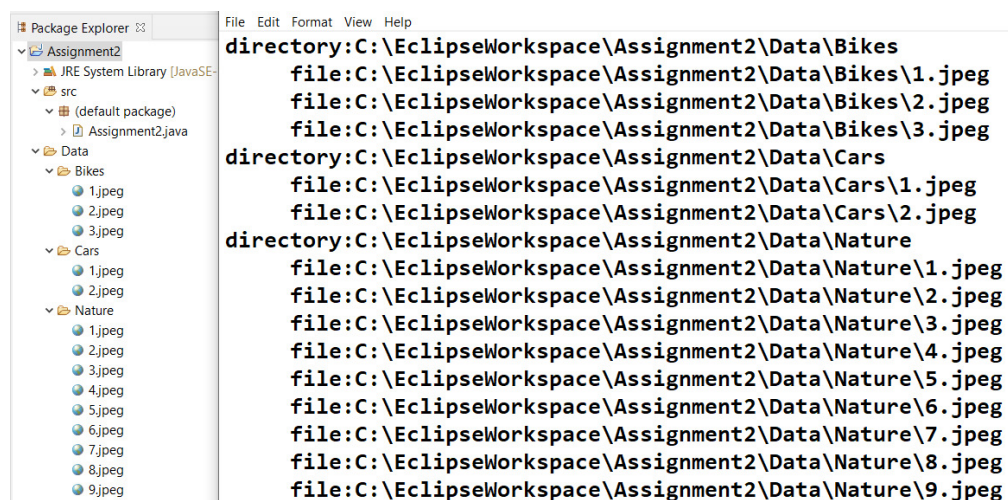


Figure 2. A sample directory structure and corresponding log file

2. The program should provide three options to users. (1) List files, (2) Process files, and (3) Exit. First option will create the "log.txt" file and second option will open "log.txt" and all the files logged in will be processed as described below. If a file named "log.txt" is already present, you must **overwrite** its contents. The third option will exit the program. You **MUST** however, close all opened files before exiting.
3. Write 2 custom exception classes **InvalidFileException** and **EmptyFolderException**. The first one responsible to make sure the log file is valid and the file paths in it are valid. The second one will be responsible for checking for empty directories. They must have constructors to allow a default error message "Error: Input file named XXX cannot be found. They should also allow for a custom message to be set as error message. All the exceptions should be logged and the files which are valid must be processed.
4. Processing phase of this application will iterate through the log file, if it exists, in the following manner:
  - a. Go through each subdirectory inside folder named "Data" and create a web page for each of the **subdirectories** with their respective name and an **.html** extension.
  - b. You should search for **jpg files** present in each **subdirectory** and put them in an html file with the subdirectory name. A sample html file is provided along with this assignment.
  - c. The **CSS** file required for presentation is also made available along with this assignment. You should simply create **HTML files** like the one provided just the file names in the <img> tag will change.
  - d. You should have **four** columns per row like the sample **html** file provided to you.
  - e. You should use the same class names as the **assignment3.html** file for presentation style.
5. Finally, here are some general information:
  - a. It may assist you greatly if you take advantage of static variables/attributes and static attributes throughout the assignment; in fact, it is not necessary to utilize other aspects such as **Inheritance**, **Polymorphism**, etc.
  - b. You must use **recursion** for creating **file structure** log.
  - c. You should **minimize opening** and **closing** the files as much as possible; a better mark will be given for that;
  - d. Do not use any external libraries or existing software to produce what is needed; that will directly **result** in a 0 mark!
  - e. Again, your program must work for any directory structure if the root directory is present in the same directory as your .java file and is named "Data". The files provided with this assignment are only one possible version and must not be considered as the **general case** when writing your **code**.

### **SUBMISSION INSTRUCTIONS**

**Submission format:** All assignment-related submissions must be adequately archived in a ZIP file using your ID(s) and last name(s) as file name. The submission itself must also contain your name(s) and student ID(s). Use your "official" name only – no abbreviations/nick names; capitalize the usual "last" name. Inappropriate submissions will be heavily penalized. If working in a group, the file name must include both IDs and last names. **IMPORTANT:** For Part II of the assignment, a demo for about 5 to 10 minutes will take place with the marker. You **must** attend the demo and be able to explain their program to the marker. The schedule of the demos will be determined and announced by the markers, and students must reserve a time slot for the demo (only one time slot per group).

Now, please read very carefully:

- If you fail to demo, a zero mark is assigned regardless of your submission.
- If you book a demo time, and do not show up, for whatever reason, you will be allowed to reschedule a second demo but a penalty of 50% will be applied.
- Failing to demo at the second appointment will result in zero marks and no more chances will be given under any conditions.

### **EVALUATION CRITERIA**

**IMPORTANT:** **Part I** must fully be submitted. Failure to submit that part will cost 50% of the total marks of the assignment!

<b>Total</b>	<b>10 pts</b>
<b>Documentations</b>	<b>1 pt</b>
<b>JavaDoc</b> documentations	1 pt
<b>Tasks</b>	<b>9 pts</b>
Task#1	2 pts
Task#2	1 pt
Task#3	2 pts
Task#4	4 pts