

# Assignment 4

## Due date

- 11.59 PM EST, April 22nd.

Submit your code as per the provided instructions. A signup sheet will be provided to you during class to setup an appointment with the TA to provide a demo of your project.

## Updates

- Sun Apr 12 19:05:12 EDT 2015: Assignment posted

## Assignment Goal

Using visitors with trees.

- You are required to use ANT for the following:
  - Compiling the code
  - running the code
  - Generating a tarball for submission
- Your code should compile and run on *remote.cs.binghamton.edu* or *bingsuns* or the *debian-pods* in the Computer Science lab in the Engineering Building.

## Project Description

### Fast Tree Visitor

- Design and develop the following 2 visitors:
  - A visitor that reads an input file, input.txt, and populates a tree data structure with all the words in the file. Use the tree data structure that will maximize performance for the second visitor. Store the words in the tree using the insert criteria for that tree data structure.
  - A visitor that counts the number of words, number of unique words, and number of characters in the tree and stores it in a file named output.txt.
  - The words from the input file should be stored/inserted in the tree using the tree property. So, if you use a BST, then you need to check *newWord < currentNodeWord*, etc.
  - It is acceptable to use data structure(s) within each tree node to store information about that node's content. However, you cannot use a separate data structure to keep track of the entire tree's data.
- In the driver code, call the two visitors and test the performance in the following manner:

```
long startTime = System.currentTimeMillis();

Start of loop N times
  declare/instantiate the tree and visitors
  code to visit with the PopulateVisitor.
  code to visit with the WordCountVisitor.
End of loop N times

long finishTime = System.currentTimeMillis();
Calculate total_time as (finishTime-startTime)/NUM_ITERATIONS.
Write the total_time value to stdout
```

- Use any whitespace as the criteria to delimit words.
- Here are three [sample input files](#).
- The following should be read from command line: input file name, output file name, and the value of NUM\_ITERATIONS.
- Use the Logger class from the previous assignment along with your own debug scheme.

## Design Requirements

- Same as previous assignments, except that javadoc is now optional.

## Code Organization

- Your directory structure should be the following:

```
lastName_firstName_assign4
---wordCount
---- README.txt
---- src
---wordCount
---- build.xml
-----driver
-----Driver
```

- (1) Validate that command line inputs for input and output file names.
- (2) Create the element(s)
- (3) Create the two visitor instances
- (4) Use the performance measurement loop given above.

```
-----util
    -----FileProcessor
        [methods to read a line and write to a file]
-----treesForStrings
    -----...
        [Classes for Tree data structures, if you define your own]
-----visitors
    -----PopulateTreeVisitor
        [Visitor class that reads words from a file and populates a tree data structure]
    -----WordCountVisitor
        [Visitor that determines the total number of words, total number of unique words, and characters in the tree]
-----other packages that you need
```

## Code Templates

- None provided for this assignment.

## Submission

- Same as Assignment-1.

## Late Submissions

- The policy for late submissions is that you will lose 10% of the grade for each day that your submission is delayed.

## Grading Guidelines

[Grading Guidelines](#).

*mgovinda at cs dot binghamton dot edu*

Back to [Design Patterns](#)