



Computer Science 3A

Practical Assignment 6

30 March 2023

Time: 30 March 2023 — 17h00

Marks: 45

Practical assignments must be uploaded to `eve.uj.ac.za` **before** 17h00 in the practical session.

Late submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student. You **must** upload your assignment to Eve **before** will be marked.

Trees are one of the most used abstract data types in Computer Science that is defined recursively as a collection of nodes is a data structure containing a value, parent and children. For this practical you are required to complete the implementation of a Tree. This Tree should conform to the implementation specifications that have been provided to you. There are a number of functions that have been removed that you should complete.

You must complete the following classes marked by:

```
//TODO: Complete
```

Please note that you should not add any additional methods in the Tree, or the TreeNode classes.

Text-based indentation based on the position in a tree

When text is structured in such a way that they can be grouped under subcategories and those subcategories may have potential subcategories the Tree abstract data type is a good data structure to store their values. In this practical the categories are the mark categories you will be getting marks for in this semester, namely your semester tests, class tests with practicals and the practical mini-project. Internal nodes represent the mark category and external nodes represent a value for its parent's mark category.

In order to solve this problem you need to complete the following functions:

1. `preOrderElementTraversal` — That traverses through the tree and returns the elements in an indent formatted String.
2. `addElementAsChild` — That adds elements to the tree using a parent position as a reference point.

3. `addChild` — The method used by `addElementAsChild` to add the element in the underlying `ArrayList`.
4. The auxiliary methods `depth` and `isInternal`.

The above functions reside in the `Tree` and `TreeNode` classes and no additional methods are required for the functioning of the assignment.

You have been provided with a `Test` class that you can use to test the execution of your solution, the results of your test program should look as similar to the output below (please note that random numbers are used, so it may look a little different).

Semester Mark Calculation

The test class stores elements of type *Mark*, which can store a name and a value. For internal nodes we store a name and a weight value. For external nodes we store only a mark value. Using the marks found in the external nodes and the weights of the mark categories (internal nodes), calculate the semester mark.

```
CSC3A 100
  ST 50
    16
    78
  MP 25
    68
  CT+PA 25
    CT 50
      55
      47
      68
    PA 50
      42
      77
      12
      42
      55
      33
      91
      3
```

Semester Mark: 53.13

The following files must be submitted to EVE:

1. *studentnumber_q6.zip*

Marksheet

- | | |
|-----------------------------------|------|
| 1. TreeNode: constructor | [3] |
| 2. TreeNode: getChildren | [2] |
| 3. TreeNode: addChild | [5] |
| 4. Tree: addElementAsChild | [5] |
| 5. Tree: preOrderElementTraversal | [10] |
| 6. Tree: depth | [5] |
| 7. Tree: isInternal | [5] |
| 8. Main: calcSM | [10] |