PROG8130
Data Structures & Algorithms for Embedded Programming

Lab 3
Binary Search Tree

Shamseddin Elmasri

selmasri@conestogac.on.ca

Objectives

- To implement a binary search tree

Procedure

A list of structures containing data and it’s associated key are defined. The keys are defined as strings and will be used to specify the location of the tree nodes (e.i: “green“ is smaller than “orange”, therefore “green” should be placed to the left of “orange”)

These structures will be fed into the binary search tree inside main() function. Data storage must satisfy the binary search tree rule. Smaller nodes go to the left.

Tree contents will then be printed out in ascending order (based on key).

Procedure (cont.)

Using the provided code as a skeleton, complete the following basic functions:

- binarySearchTree_AddNode()
- binarySearchTree_PrintTreeInOrder()

- Note: When printing out nodes. Make sure to print out both the key and the associated data.

Submission

- Submit your completed program to eConestoga.
- Program must be in a single (.c) source file that can be complied on its own.
- Any number of submissions will be accepted, but only the latest one before the deadline will be evaluated.

Evaluation

- Your program will be evaluated for the following criteria:
  - Correct functionality.
  - Coding style, proper indentation, comments.
  - Compilation without any warnings or errors.
  - Proper error handling. (e.g. handling the case where malloc() returns NULL)
  - Marks will be deducted for:
    - Extra debug left in program after submission
    - Extra unused elements in data structures

Basic Outcomes (55% to 79%)

- binarySearchTree_AddNode() and binarySearchTree_PrintTreeInOrder() functions operate as per procedure.

Advanced Outcomes (80% to 100%)

1. Basic outcomes fully met
2. Code implemented without recursion.