## CS 201, Fall 2022

## Homework 3

**DUE: November 25, Friday @23:59**

**Please check the submission rules towards the end of the document.**

**Points will be deducted in case of a violation of these rules!**

**Description:** In this assignment you will write a C++ program that takes a set of integer values as input, and constructs a **binary search tree**. Then, it will print out the entered values by traversing the constructed tree with **preorder**, **inorder**, and **postorder** traversal strategies.

**1) Input and Expected Output**

The program will continuously ask for positive numbers and construct the tree until -2 is entered. After -2 is entered, the program will continuously ask for positive numbers and remove such numbers from the tree until -1 is entered. When -1 is entered, it will print out the contents of the tree with preorder, inorder, and postorder traversals. Duplicate values, if exist, will be ignored. Missing values for removal will also be ignored. A sample output is shown below.

Enter a set of numbers (-2 to remove numbers, -1 to stop):

**20**

**12**

**18**

**23**

**7**

**19**

**21**

**32**

**5**

**11**

**35**

**-2**

**12**

**11**

**7**

**35**

**100**

**-1**

preorder: 20 18 5 19 23 21 32

inorder: 5 18 19 20 21 23 32

postorder: 5 19 18 21 32 23 20

**2) The Design**

This assignment will involve the implementation of two classes and a *cpp* file which includes the *main* method of the program. The *header files* of the classes *will be provided* to you. You are free to modify/extend the classes with new type of attributes and operations; however, the public operations should be in any case supported by your overall design as depicted below. Hereby, public operations are annotated with a “+” in the diagram, whereas private operations are annotated with “-”.



The *main* method creates a *BinarySearchTree* object, receives the input numbers from the user, and inserts these elements to the tree by using the *insert* method. After the user enters -2, it receives the input numbers from the user, and deletes these elements from the tree by using the *remove* method. Then, after the user enters -1, it calls the *printPreorder*, *printInorder* and *printPostOrder* methods of the tree object. These methods print out all the elements remained in the tree after insertion/removal while traversing the nodes in preorder, inorder and postorder, respectively.

All the public methods of the *BinarySearchTree* class make calls to internal, private methods that are *recursively* called. For instance, the *insert* method that is called by the *main* method, in turn makes a call to the private *insert* method by specifying *root* as the *BinaryTreeNode pointer* argument. This private method can create a new node with the specified element and assign this newly created node to the root if the root is null. Otherwise, if the root is not null, it can make a recursive call with the left or right child node of the root as the argument. Similarly, all the other private methods are initially called with the root node as the argument. The private method *deleteNodes* is supposed to be called by the *destructor* to delete all the nodes in the tree in a bottom-up manner (i.e., post order traversal). This method will also be called with the root node as the argument and it will traverse the tree recursively to delete every visited node after its descendants are deleted.

**3) Submission**

Youwill submit this homework via the LMS system. You should follow the file-naming conventions and guidelines below.

* You should submit your source files as a **ZIP** archive file (**NOT** RAR or other formats). The name of the file should be in format “**<USER-ID>\_hw<HOMEWORK-NR>.zip**”. For example, if your username is vy1043, then the name of the submitted file should be “vy1043\_hw3.zip”. Pay attention that all the letters are in lower-case. ZIP archive is supposed to contain **just the source files**, no folders are allowed by any means.
* The contents of the ZIP file should be as follows:
  + **hmw3.cpp** (includes the *main* function)
  + **BinarySearchTree.h** (BinarySearchTree class definition)
  + **BinarySearchTree.cpp** (BinarySearchTree class implementation)
  + **BinaryTreeNode.h** (BinaryTreeNode class definition)
  + **BinaryTreeNode.cpp** (BinaryTreeNode class implementation)
* Late submissions and C++ files that do not compile are **not** accepted.
* You can resubmit your homework (until the deadline) if you need to.
* Make sure that your program does **not** include commands specific to a development environment, e.g., *system(“pause”)* or *#pragma once* in Visual Studio.