Computer Science Department
San Francisco State University
CSC 340
Spring 2016

Assignment 6 - Templates and Trees

Due Date
Wednesday, April 27, at midnight.

Overview
The purpose of this project is to implement a template **`Tree`** class that will be able to hold instances of our **`BudgetEnvelope`**. You are provided the **`Tree`** implementation we completed in class, and must change the implementation to allow multiple types to be stored in the **`Node`**s.

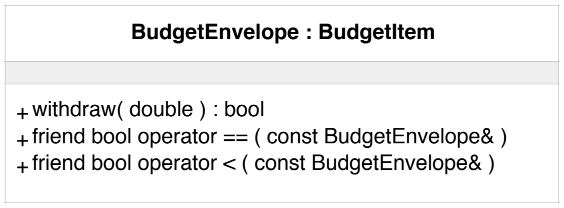
Submission
See the submission guidelines posted on iLearn.

Requirements
1. Implement the **`Tree`** class as a template that can be used to store a month's worth of **`BudgetEnvelope`**s.
    1.1. Remove references to **`int`**, and combine into one file (remember that template classes are defined in the header file only!)
    1.2. Implement the find method that returns the **`Node`** value (since we will need to perform operations on **`BudgetEnvelope`**s)
2. Add the following overloaded operators to the **`BudgetEnvelope`** class so that insert, search, and find functionality will work for our custom type:
    2.1. Overloaded **`< operator`** (used in insert and search)
    2.2. Overloaded **`== operator`** (used in search)
3. Write a test driver
    3.1. Note that this is the first time you are being asked to do this - it's good practice to figure out how to test and validate your code!! Feel free to ask questions on the forum!

Resources
The makefile for this assignment can be found at https://gist.github.com/jrob8577/2aa923ae477ef63debdc3cc53c4fb027 (the same as for assignment 5 - template classes are only included via header include, and you may use main.cpp to write your driver).

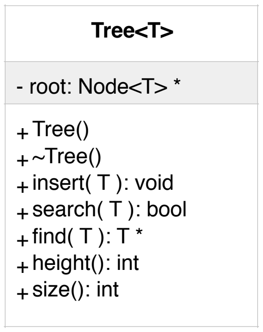
The tree implementation completed in class can be found at https://github.com/jrob8577/tree-csc340. Note that this may not include all of the behavior necessary for this assignment, carefully read the requirements!

Appendix A: UML Diagrams



BudgetEnvelope : BudgetItem

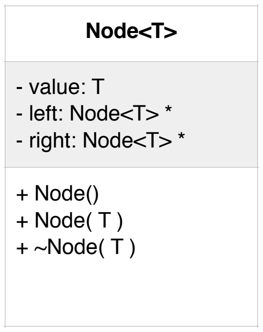
+ withdraw(double) : bool
+ friend bool operator == (const BudgetEnvelope&)
+ friend bool operator < (const BudgetEnvelope&)



Tree<T>

- root: Node<T> *

+ Tree()
+ ~Tree()
+ insert(T): void
+ search(T): bool
+ find(T): T *
+ height(): int
+ size(): int



Node<T>

- value: T
- left: Node<T> *
- right: Node<T> *

+ Node()
+ Node(T)
+ ~Node(T)