

## **CSCI124/MCS9124 Applied Programming Autumn 2015**

### **Lab 12 (1 mark)**

Due at 11:59pm on Thursday, June 4th.

#### **Aim:**

To use traversals and function pointers to perform operations on nodes in a tree

#### **Task:**

To use the code for the class `BinaryTree` provided in the files `binarytree.h` and `binarytree.cpp`, we have to supply two things: a file called `datatype.h` which defines the type `T` to go into the tree; and declares functions

```
int cmp(const T&, const T&);
void copy(T&, const T&);
void print(const T&);
```

For the purposes of this lab, just make `T` an `int`.

The implementations of the functions can go in `datatype.cpp`.

First, write a main program in `main.cpp` which creates a binary search tree and fills it with 20 random values between 0 and 99 inclusive and then prints the values in order.

Write a function `void TwoTimes(const T&)` in `main.cpp` which, instead of printing the value of the given integer, prints twice its value. We will now write methods in the search tree to apply this function to every node in an in order traversal:

- Write a prototype for the method `applyToAll` in the `BinaryTree` class just after `printAll`. It should return `void`, and should take a pointer to a function called `func` which returns `void` and takes a `const T&`. Like `printAll`, it should be a `const` method.
- Write a prototype for the method `inorderApply` in the `BinaryTree` class just after `inorderPrint`. It should return `void`, and should take a `np_ptr` called `cur`, a to a function called `func` which returns `void` and takes a `const T&`. It should be also a `const` method.
- Now write the implementations for the above methods.
  - `applyToAll` is similar to `printAll`, but instead of calling `inorderPrint`, it calls `inorderApply`.
  - `inorderApply` is similar to `inorderPrint` except instead of printing, it calls `func`, passing it the data for the current node. Note you will also need to modify the function called

for the left and right subtrees appropriately.

- In your main program call `applyToAll` passing it the function `TwoTimes`. Run your program, it should print double the value of each node in sorted order. This is not a very useful function.
- In `main.cpp`, declare a global `int sum` and set it to 0. Write another function in `main` with the same prototype as `TwoTimes` but a different name, which adds the given integer to `sum`. Extend your main program to print the sum of all numbers in the tree.

Finally, notice that the only print function available is an inorder print.

- Write a public `preorderPrintAll` method for the class. Maintain the same style by implementing it using a private `preorderPrint` method.

### Submission:

```
$ submit -c csci124 -a lab12 -u <username> binarytree.cpp  
binarytree.h datatype.h main.cpp
```