**<Data Structure Assignment 3>**

2016043654 경제금융학부 이창모

/\* **Task-1**: Complete the code for ‘inOrder’, ‘postOrder’, and ‘levelOrder’. \*/

template <class T>

void inOrder(binaryTreeNode<T>\* t)

{

if (t != NULL)

{

inOrder (t->leftChild);

visit(t);

inOrder (t->rightChild;

}

}

template <class T>

void postOrder(binaryTreeNode<T>\* t)

{

if (t != NULL)

{

postOrder (t->leftChild

postOrder (t->rightChild);

visit(t);

}

}

template <class T>

void levelOrder(binaryTreeNode<T>\* t)

{

arrayQueue<binaryTreeNode<T>\*> q;

while (t)

{

visit(t);

if (t->leftChild)

q.push(t->leftChild);

if (t->rightChild)

q.push(t->rightChild);

try { t = q.front(); }

catch (queueEmpty empty)

{

cout << endl;

empty.outputMessage();

//cout << "\nAll elements are printed" << endl;

}

try { q.pop(); }

catch (queueEmpty)

{

return;

}

}

}

/\* **Task-2**: Complete the code for ‘main’. The input tree and output format are described in the comment. Refer to the sample section of the code for more information. \*/

int main(void)

{

binaryTreeNode<int> \* c,\* d,\* e, \* a, \* b;

c = new binaryTreeNode<int>(3);

d = new binaryTreeNode<int>(4);

e = new binaryTreeNode<int>(5);

b = new binaryTreeNode<int>(2, d, e);

a = new binaryTreeNode<int>(1, b, c);

cout << "Inorder: ";

inOrder(a);

cout << endl;

cout << "PreOrder: ";

preOrder(a);

cout << endl;

cout << "PostOrder: ";

postOrder(a);

cout << endl;

cout << "LevelOrder: ";

levelOrder(a);

cout << endl;

return 0;

}

/\* **Task-3**. Complete your Makefile to compile your code. \*/

make

sudo make install

CC = g++

CFLAGS = -g -Wall

LFLAGS = -D\_GLIBCXX\_ASSERTIONS

TARGET = app.out

all: $(TARGET)

$(TARGET): $(TARGET).cpp

$(CC) $(CFLAGS) -o $(TARGET) $(TARGET).cpp

clean:

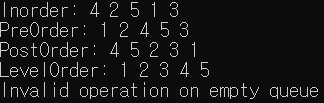
$(RM) $(TARGET)

app.out: binaryTreeTraversal.o

gcc -o app.out binaryTreeTraversal.o

binaryTreeTraversal.o: arrayQueue.h binaryTree.h binaryTreeNode.h myExceptions.h queue.h binaryTreeTraversal.cpp

gcc -c -o binaryTreeTraversal.o binaryTreeTraversal.cpp

make