CSCI 321 Computer Science III Fall 2018

Assignment 2

1. Textbook (Goodrich 6th edition), Chapter 8 Pg. 351, Problem R-8.22
2. Textbook (Goodrich 6th edition), Chapter 2 Pg. 351, Problem R-8.15. Hint: Review the definition of the level numbering and use this definition when passing information from a node to its children.
3. Implement a binary tree using the array-based representation shown in Chapter Slide 33. Code in Java with main class. Use the binary tree in Slide 33 as an input instance. Attach your code and screenshot of the output. See next page for example code (details of some methods are missing.)

// JAVA implementation of tree using array

// numbering starting from 0 to n-1.

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class Tree {

public static void main(String[] args)

{

Array\_imp obj = new Array\_imp();

obj.Root("A");

obj.set\_Left("B", 0);

obj.set\_Right("C", 0);

obj.set\_Left("D", 1);

obj.set\_Right("E", 1);

obj.set\_Left("F", 2);

obj.print\_Tree();

/\*Code above is an example of an instance. You need to input the tree shown in slide 33\*/

}

}

class Array\_imp {

static int root = 0;

static String[] str = new String[10];

/\*create root\*/

public void Root(String key)

{

str[0] = key;

}

/\*create left son of root\*/

public void set\_Left(String key, int root)

{

/\*need your code here\*/

if(str[root] == null){

System.out.printf("Can't set child at %d, no parent found\n",t);

}else{

str[t] = key;

}

}

/\*create right son of root\*/

public void set\_Right(String key, int root)

{

/\*need your code here\*/

if(str[root] == null){

System.out.printf("Can't set child at %d, no parent found\n",t);

}else{

str[t] = key;

}

}

public void print\_Tree()

{

for (int i = 0; i < 10; i++) {

if (str[i] != null)

System.out.print(str[i]);

else

System.out.print("-");

}

}

}