# Homework Turnin

**Account:** 6G_06 (rgalanos@fcps.edu)
**Section:** 6G
**Course:** TJHSST APCS 2016-17
**Assignment:** 11-01

**Receipt ID:** 13cc709efaef8ee91194616768b58067

**Warning:** Your turnin is 1 day late. Assignment 11-01 was due Wednesday, May 24, 2017, 12:00 AM.

Execution failed with return code 1 (general error). (Expected for JUnit when any test fails.)

**Warning:** Your program failed to compile :

```
deHuffman_teacher.java:81: error: duplicate class: TreeNode
class TreeNode
^
1 error
```

Please correct your file(s), go back, and try to submit again. If you do not correct this problem, you are likely to lose a large number of points on the assignment. Please **contact your TA** if you are not sure why your code is not compiling successfully.

## Turnin Failed! (See above)

There were some problems with your turnin. Please look at the messages above, fix the problems, then Go Back and try your turnin again.

GradeIt has a copy of your submission, but we believe that you will want to fix the problems with your submission by resubmitting a fixed version of your code by the due date.

We have received the following file(s):

deHuffman.java (2921 bytes)

```java
// Name:              Date:
import java.util.Scanner;
import java.io.*;
public class deHuffman
{
   public static void main(String[] args) throws IOException
   {
      Scanner keyboard = new Scanner(System.in);
      System.out.print("\nWhat binary message (middle part)? ");
      String middlePart = keyboard.next();
      Scanner sc = new Scanner(new File("message."+middlePart+".txt"));
      String binaryCode = sc.next();
      Scanner huffLines = new Scanner(new File("scheme."+middlePart+".txt"));

      TreeNode root = huffmanTree(huffLines);
      String message = dehuff(binaryCode, root);
      System.out.println(message);

      sc.close();
      huffLines.close();
   }
   public static TreeNode huffmanTree(Scanner huffLines)
   {
      TreeNode tree = new TreeNode(null);
      while(huffLines.hasNext())
      {
         TreeNode child = tree;
         String str = huffLines.nextLine();
         for(int i=1;i<str.length();i++)
         {
            if(str.charAt(i)=='0')
            {
               if(child.getLeft()==null)
                  child.setLeft(new TreeNode(null));
               child = child.getLeft();
            }
            else
            {
               if(child.getRight()==null)
                  child.setRight(new TreeNode(null));
               child = child.getRight();
            }
         }
         child.setValue(str.charAt(0)+"");
      }
      return tree;
   }
   public static String dehuff(String text, TreeNode root)
   {
      String str = "";
      int i=0;
      while(i<text.length())
      {
         TreeNode temp = root;
         int length = str.length();
         while(str.length()==length)
         {
            if(temp.getValue()!=null)
               str+=temp.getValue();
            else if(text.charAt(i)=='0')
            {
               temp = temp.getLeft();
               i++;
            }
            else
            {
               temp = temp.getRight();
               i++;
            }
         }
      }
      return str;
   }
}

 /* TreeNode class for the AP Exams */
class TreeNode
{
   private Object value;
   private TreeNode left, right;

   public TreeNode(Object initValue)
   {
      value = initValue;
      left = null;
      right = null;
   }

   public TreeNode(Object initValue, TreeNode initLeft, TreeNode initRight)
   {
      value = initValue;
      left = initLeft;
      right = initRight;
   }

   public Object getValue()
   {
      return value;
   }

   public TreeNode getLeft()
   {
      return left;
   }

   public TreeNode getRight()
   {
      return right;
   }

   public void setValue(Object theNewValue)
   {
      value = theNewValue;
   }

   public void setLeft(TreeNode theNewLeft)
   {
      left = theNewLeft;
   }

   public void setRight(TreeNode theNewRight)
   {
      right = theNewRight;
   }
}

```