ITCS 1213
February 16, Monday

General:

Topics:

Reading data from a file and printing to a file

String class

String objects are immutable

StringBuilder class

Purpose

constructor methods

a look at the API documentation to see what methods are available

examples

THPT2

read requirements

Questions about the requirements

Class ReadingFile - ReadingFile.java

```
1  /**
2   * This is an example of how to create a File object
3   * to send to the Scanner class constructor. This
4   * allows your program to read data from a file.
5   *
6   * @author: L. Lehmann
7   * @version 2-11-2015
8   */
9  import java.util.*;
10 import java.io.*;
11
12 public class ReadingFile
13 {
14     //the method that opens the file must declare this throws clause
15     public static void main(String[ ] args) throws IOException
16     {
17         String fileName;
18         Scanner   keyboard = new Scanner(System.in);//keyboard input
19         Scanner   inputFile;//will hold a Scanner reference to a File str
   eam
20
21         String word;
22         int age;
23
24         System.out.println("Enter the name of the file: ");
25         fileName = keyboard.nextLine( ) ;
26         //I am creating a file here in the parameter list
27         inputFile = new Scanner(new File(fileName));
28         //hasNext()returns true if there is more data,false if no more da
   ta
29         while(inputFile.hasNext())
30         {
31
32             word = inputFile.next( );//the next( ) and nextInt( ) methods
33             age = inputFile.nextInt( ); //are from the Scanner class
34
35             System.out.println(word + "   "  + age);
36         }//end of while loop
37
38         inputFile.close( );//close the file stream when you are done
39      }//end of main
40  }
41
```

Class FileOperations - FileOperations

```
1  /**
2   * This is an example of how to write data
3   * to an output file using the PrintWriter class
4   *
5   * @author L. Lehmann
6   * @version 2-11-2015
7   */
8  import java.io.*;
9  import java.util.*;
10
11 public class FileOperations
12 {
13      public static void main(String [ ] args) throws IOException
14      {
15          PrintWriter output = new PrintWriter("c:\\mydata.txt");
16          output.println("Today is Wednesday.");
17          output.println("Tomorrow is Thursday.");
18          output.println("I can't wait for lab to start");
19          output.close( );//close the file when done writing
20          //print a message to the screen so user knows job is done
21          System.out.println("Printing complete. ");
22      }
23 }
24
25
26
```

Class Replace - StringBuilder Example - Part 3

```
1
2   /**
3    * Write a description of class Replace here.
4    *
5    * @author L. Lehmann
6    * @version 2-16-2015
7    */
8   public class Replace
9   {
10      public static void main(String[ ] args)
11      {
12          StringBuilder strb;
13          int startPos;
14          String replaceWith = new String ("beautful");
15
16          strb = new StringBuilder("Today is a wonderful day");
17          System.out.println(strb);
18          startPos = strb.indexOf("wonderful");
19
20          if (startPos != -1)
21              strb.replace(startPos, startPos + "wonderful".length(), "beau
    tiful");
22
23         System.out.println(strb);
24       }
25
26      }
27
28
```

Class Driver - String Manipulation

```
1  /**
2   * Example of using the StringBuilder class
3   * The StringBuilder class is used to manipulate
4   * strings of characters.
5   */
6  import java.util.*;
7  import java.io.*;
8
9  public class Driver
10 {
11  public static void main(String[ ] args) throws IOException
12  {
13      File inText = new File ("c:\\lines of text.txt");
14      Scanner input = new Scanner(inText);
15
16
17      StringBuilder strToWork;
18
19      while (input.hasNext( ))
20      {
21          strToWork = new StringBuilder(input.nextLine());
22
23          for(int i = 0; i < strToWork.length( );i++ )
24          {
25              if(strToWork.charAt(i) == 'a' ||strToWork.charAt(i) == 'e'
26              || strToWork.charAt(i) == 'i' ||
27              strToWork.charAt(i) == 'o'|| strToWork.charAt(i) == 'u')
28              {
29
30                    strToWork.replace(i, i+1, "x");
31              }
32          }
33          System.out.println(strToWork);
34      }
35    }
36 }
37
38
```

ITCS 1213
Take Home Programming Test 2
Spring 2015

Due: Sunday, March 1 11:55pm to your lab Moodle site

String Problem

For each numerical value 0, 1, 2, ...9 (0 <= NUMBER <= 9), embedded in a sentence, convert that value to its equivalent English text. Print the converted sentence both to the screen and to an output file.

Your input file consists of a variable number of records. Each record is a sentence of length <= 80 characters. More than one numerical value in the given range may appear in a sentence. You must deal with upper and lower case issues. If a line begins with a single digit, write that digit as a word starting with an uppercase letter. See the examples below.

Examples:

Input Record:
3 foxes were chasing 5 rabbits and 10 ducks.
Output Record:
Three foxes were chasing five rabbits and 10 ducks.

Input Record:
I used 3 eggs out of the 12 for the cake.
Output Record:
I used three eggs out of the 12 for the cake.

Input Record:
1 picture is worth 1000 words.
Output Record:
One picture is worth 1000 words.

Input Record:
There are 260 students enrolled in Java.
Output Record:
There are 260 students enrolled in Java.

Create the following as an input file for testing.

The 8 eggs were separated into 3 groups.
5 boys and 7 girls were present.
He was 1 hour and 5 minutes late.
She ate 3 dozen doughnuts!
4 dogs were chasing 3 cats.
The captain said, "This is the 0 hour".
I tried to call you 9 times today; Ann tried 6 times!!
The 12 firemen worked quickly.

Prompt the user for the name of the input file. Name your output file "outSentence.txt" . Save the output file in the same directory as your code to make grading on different systems easier.

More details:

Create two class files. One class is the Converter. It has a StringBuilder field for the original sentence and a String field for the converted sentence.

The constructor will call a method to convert the original sentence.

You are to use only the methods of the StringBuilder class for the conversion. These are the only methods of the String and StringBuilder class you are permitted to use:

StringBuilder:
- constructor s
- charAt( )
- indexOf( )
- length( )
- replace( )

String:
- constructors
- charAt( )
- length( )

You will also have to use methods from the Character class.

The second class is the driver class The driver will contain the main( ) method. The main( ) method will open a file for input. Read the file line by line and send each line to the Converter class. The main( ) method will call the get method( ) of the Converter class to get the converted string, and print this string both to the screen and to a file.

Documentation standards:

A flowerbox with Javadoc at the top of each class. Include the file name, the date, your lab section, and a description of the class.

On top of each method, use Javadoc comments for the name of the method, its purpose, describe the incoming parameters and the return value. Use meaningful field and variable names. Use inline comments to document your logic. Follow all naming conventions.

Cheating: Cheating will be punished according to the standards set in the course syllabus. You are NOT to get help from another student, from the CCI Student Center, from teaching assistants, from relatives or friends, or from the Internet on this test. This is an individual test.

I will post similar problems that you can ask for help with.