Lab 4: String

Part 1: Main Concepts:

1. Matching:
2. UNICODE character set 4
3. substring method of String class 6
4. indexOf method of String class 1
5. StringBuilder class 2
6. String class 5
7. Token 3

(1) ~~the index within this string of the first occurrence of the specified character or -1 if the character does not occur.~~

~~(2) This class is designed for use as a drop-in replacement for StringBuffer in places where the string buffer was being used by a single thread~~

~~(3) individual words and punctuation marks~~

(4) ~~computing industry standard designed to consistently and uniquely encode characters used in written languages throughout the world~~

~~(5) this class includes methods for examining individual characters of the sequence, for comparing strings, for searching strings, for extracting substrings, and for creating a copy of a string with all characters translated to uppercase or to lowercase~~

~~(6) two variants and returns a new string that is a subset of this string~~

1. Determine the results of the code:
2. Describe what this program is doing:

Scanner scanner = new Scanner(System.in);

System.out.println("Please enter first string:");

String firstString = scanner.nextLine();

System.out.println("Please enter second string:");

String secondString = scanner.nextLine();

int value = firstString.compareTo(secondString);

System.out.println("\nCompare Result:");

if (value == 0)

System.out.printf("%s == %s\n", firstString, secondString);

else if (value > 0)

System.out.printf("%s > %s\n", firstString, secondString);

else

System.out.printf("%s < %s\n", firstString, secondString);

This code gets two strings from the user. Then it compares the strings and returns the number of charectors that are different. The “if(value == 0)” means the strings are identical. The “else if (value > 0)” means that string 1 is greater value then string 2

Part 2: Implementation String Tokenizer, Regular Expression

**Step 1:** Under your Lab project, create Lab4 package. Select Lab4 package and create the class PhoneTest, in which you write an application that takes as inputs a telephone number and check the format of this phone number. A valid telephone number should be in the form *???<any spaces> ???<any space>????* (note that each ‘?’ represents a single digit, the first digit should be non-zero). Your program should tell the user if the user’s phone is in valid format or not.

Hint:

* you should create a regular expression that represents the format of a valid phone number
* you can use matches method (of String class) to compare the user’s phone number with the regular expression that you have created.

**Step 2:** Select Lab4 package again and create a class TokenTest in which a user enters a line of text. Your program will tokenize this line and display all the tokens in reverse order. Use space character as delimiter.

Hint:

* You can use Scanner (as shown in the book) or JOptionPane for getting inputs from users
* You should use either StringTokenizer class or split method in String class to do tokenizing task.
* You should compute the number of tokens first and dynamically allocate memory for the array of length = number of tokens.

Submission: submit your answers in part 1 in \*.doc file and your \*.java in Lab 4 dropbox by the due date.