Matt Matuk

CSIT – 211

28 February 2015

Lab 1

**Programming Project PP 12.2**

-----**SPEC**-----

Design and implement an application that reads a sentence from the user and prints the sentence with the characters of each word backwards. Use a stack to reverse the characters of each word.

**SCRUM**

* Class Reverse
  + Method Main
* Class ReverseOrder
  + Variables
    - String sen
    - Stack reverse
  + Method reverseSentance()
  + Method toString()

**Class: Reverse**

1. This class contains the main method of the application.
2. This class will create a ReverseOrder object and call the reverseSentance() method to reverse the order of each word in a sentence.
3. Make sure anything that is printed looks nicely formatted.

**Import**

* Scanner

**Process**

* Variables
  + Scanner scan
  + String reverse
* Method main()
  + Create a new ReverseOrder object
  + Create a new scanner.
  + Print some intro statements for the user of the application
  + Print “Please enter a sentence to reverse the order of each word.”
  + reverse equals the return value of the method reverseSentance(scan.nextline). Pass the method sentence the user entered.
    - When scanning for the users sentence. Make sure the user can only enter a max of 256 characters. If they enter more, loop thru until they enter the correct amount.
  + Print “Here is the sentence with the words reversed.”
  + Print the string reverse.
  + Print some ending statements.
  + Close the scanner

**Class: ReverseOrder**

1. This class will accept a string and return that string with the words backwards.
2. The class will use a stack to reverse the order of the words.
3. All variables are protected unless otherwise noted
4. All methods are public and void unless otherwise noted

**Import**

* Scanner
* Stack

**Process**

* Variables
  + String sen
  + Stack<String> reverse
* Method constructor()
  + Does nothing
* Method String reverseSentance(String sentence)
  + Set sen equal to the parameter sentence
  + Create a new string stack result
  + Create a new string reverse = “”
  + Create a new scanner scan to which will can the string sen
  + While scan hasNext for the string sen
    - If the character is equal to a space
      * Pop each item in the reverse stack and append the value to the string result
    - Else
      * Add the character to the stack reverse and continue the loop until no characters are left
  + Return result

**CODE**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Reverse.java Matt Matuk

// CSIT 211 Project 12.2 pg 463

// 1. This class contains the main method of the application.

// 2. This class will create a ReverseOrder object and call

// the reverseSentance() method to reverse the order of each

// word in a sentence.

// 3. Make sure anything that is printed looks nicely formatted.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**import** java.util.Scanner;

**public** **class** Reverse

{

**public** **static** **void** main(String[] args)

{

Scanner scan = **new** Scanner(System.***in***);

ReverseOrder reverOrder = **new** ReverseOrder();

String reverse;

String in;

System.***out***.println("Welcome to my application. \n"

+ "Today you will enter a sentance and then I "

+ "will print that sentance with each word "

+ "backwords.");

System.***out***.println();

// only allows 256 character sentance max.

**do**

{

System.***out***.println("Please enter a sentance to reverse "

+ "the order of each word. Max charater is 256.");

in = scan.nextLine();

**if** (in.length() > 256)

{

System.***out***.println("Error. Please enter less "

+ "than 256 Charaters.");

System.***out***.println();

}

} **while**(in.length() > 256);

reverse = reverOrder.reverseSentance(in);

System.***out***.println();

System.***out***.println("Here is the sentance with each word "

+ "backwords.");

System.***out***.println(reverse);

System.***out***.println();

System.***out***.println("Thank you for using my application "

+ "today.");

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ReverseOrder.java Matt Matuk

// CSIT 211 Project 10.7 pg 463

// 1. This class will accept a string and return that string with

// the words backwards.

// 2. The class will use a stack to reverse the order of the

// words.

// 3. All variables are protected unless otherwise noted

// 4. All methods are public and void unless otherwise noted

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**import** java.util.Stack;

**public** **class** ReverseOrder

{

**protected** String sen;

**protected** Stack<String> reverse;

**public** ReverseOrder()

{

}

//-------------------------------------------------------------------

// This method is where the passed string is reverse word by word.

//-------------------------------------------------------------------

**public** String reverseSentance(String sentance)

{

sen = sentance;

reverse = **new** Stack<String>();

String result = "";

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Debug\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//System.out.println(sen);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// loops thru each character in the string

**for** (**int** count = 0; count<sen.length(); count++)

{

// if the character equals a space, then the stack is poped

// until its empty.

**if** (sen.substring(count,count+1).equalsIgnoreCase(" "))

{

result = result + pop(reverse.size(), **true**);

}

// each character is added to the stack until a space is

// reached, then the stack is poped to the result string.

**else**

{

reverse.push(sen.substring(count,count+1));

}

}

// this is for the end of the sentance to clear the last word.

result = result + pop(reverse.size(),**false**);

**return** result;

}

//-------------------------------------------------------------------

// This method pops the stack clear by looping the size of teh stack

//-------------------------------------------------------------------

**private** String pop(**int** times, **boolean** space)

{

String result = "";

**for** (**int** num = 0; num<times; num++)

{

result = result + reverse.pop();

}

**if** (space)

{

result = result + " ";

}

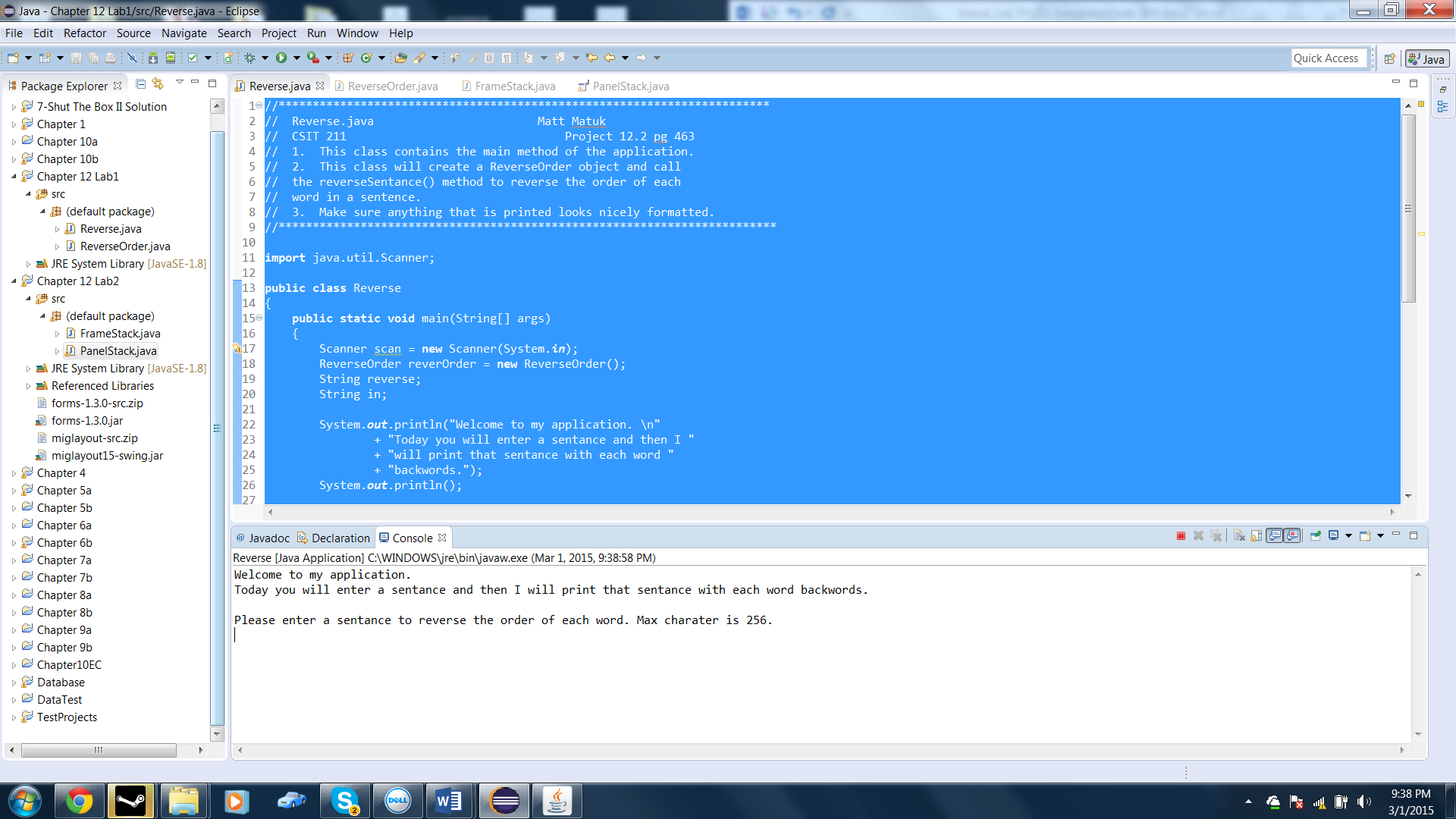
**return** result;

}

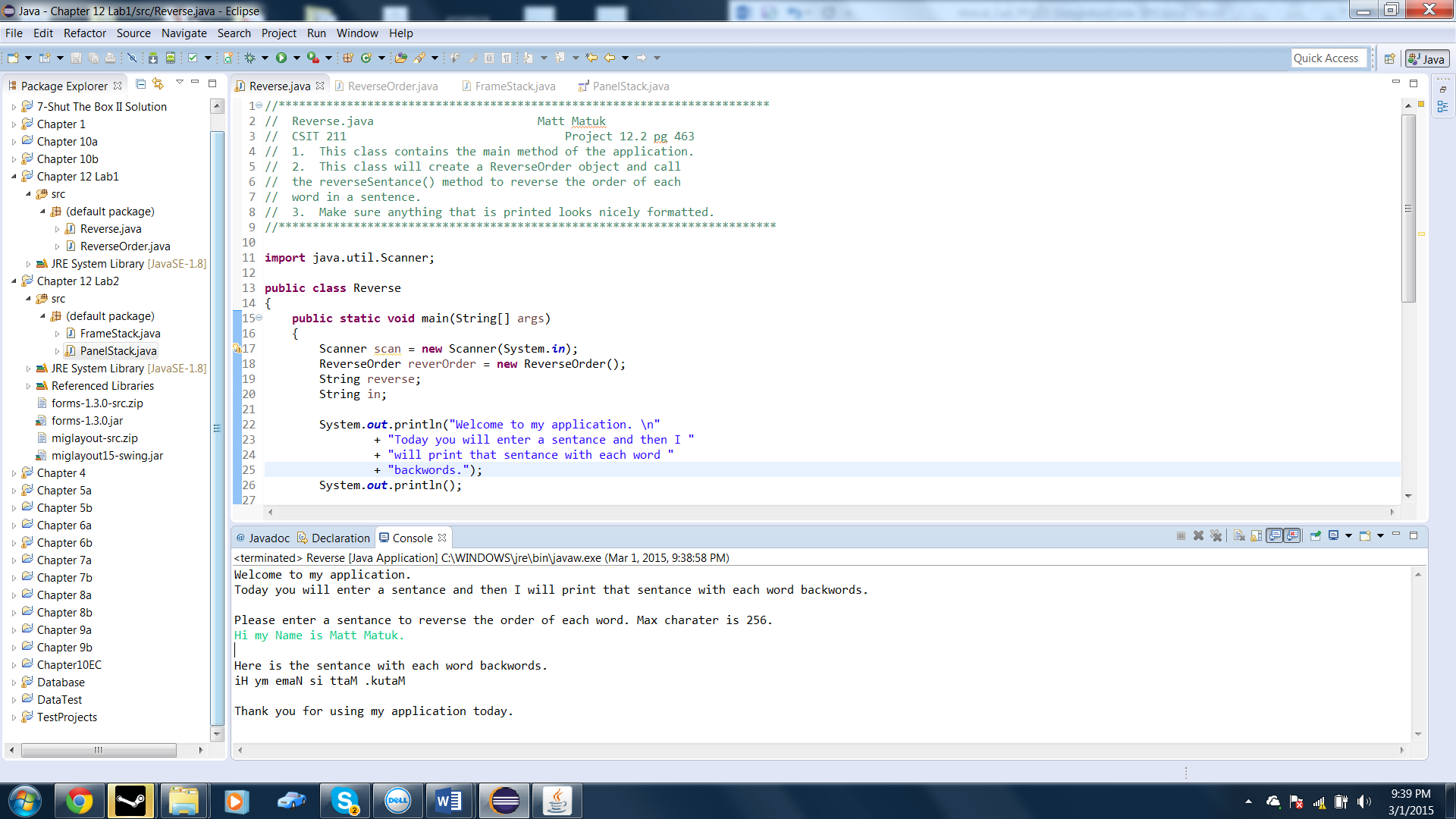
}

**OUTPUT SCREEN**

* Starting program.



* After entering a sentence.



* Error message when entering more than 256 characters. Note: some characters are not shown.

