|  |  |
| --- | --- |
| **Exposure Java** | **Lab 08** |
| **The "Palindrome" Program** | **80 & 100 Point Versions** |
| **Assignment Purpose:**  This program requires knowledge and manipulation of Java **String** objects.  Do not use the **charAt** method for this program. | |

Write a program that determines if an entered string is a **Palindrome**. True palindromesare strings of characters that read the same backward as forward. This does mean all characters, including spaces and punctuations. Examples of palindromes are:

**MADAM, RACECAR, BOB, HANNAH, CIVIC, KAYAK, LEVEL, REVIVER**

|  |  |
| --- | --- |
| **Lab 08 Student Version** | **Do not copy this file, which is provided.** |
| // Lab08st.java // The "Palindrome" Program // This is the student starting file.  import java.util.Scanner;  public class Lab08st {  public static void main (String args[])  {  System.out.println("\nLab08v80\n");  Scanner input = new Scanner(System.in);  boolean notFinished = false;  do  {  System.out.print("Enter a string ===>> ");  String str = input.nextLine();  System.out.println();  System.out.println("Entered String: " + str);  System.out.println("Palindrome: " + Palindrome.isPal(str));  System.out.println("Almost Palindrome: " + Palindrome.almostPal(str));   System.out.print("Do you wish to repeat this program [Y/N]? ===>> ");  String repeat = input.nextLine();  notFinished = (repeat.equals("Y")) || (repeat.equals("y"));  System.out.println();  }  while (notFinished);  } }  class Palindrome {  /\*  \* Precondition: s is an arbitrary String.  \* Postcondition: The value of true is returned if s is a Palindrome, false otherwise.  \* Note: This method is required for both the 80 point and the 100 point.  \*/  public static boolean isPal(String s)  {  return true; // This statement is provided to allow initial compiling.  }     /\*  \* Precondition: s is a String of one character.  \* Postcondition: The value of true is returned if s is a letter and false otherwise.  \* Note: >>>>> This method is only completed for the 100 point version <<<<<  \*/  private static boolean isLetter(String letter)  {  return true; // This statement is provided to allow initial compiling.  }    /\*  \* Precondition: s is an arbitrary String.  \* Postcondition: All non-letter characters are removed from s.  \* Note: This method is only completed for the 100 point version.  private static String purge(String s)  {  return ""; // This statement is provided to allow initial compiling.  }    /\*  \* Precondition: s is an arbitrary String.  \* Postcondition: After purging all non-letter characters from s,  \* the value of true is returned if the resulting String is a  \* Palindrome, false otherwise.  \* Note: This method is only completed for the 100 point version.  \*/  public static boolean almostPal(String s)  {  return true; // This statement is provided to allow initial compiling.  }  } | |

**80 Point Version Specifics**

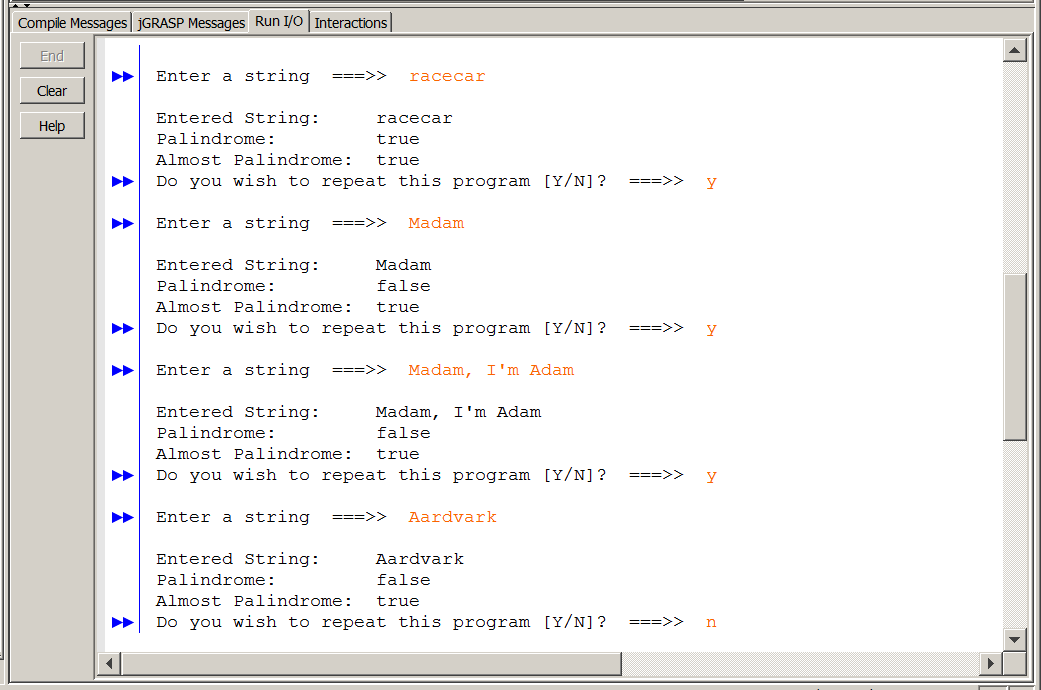
The main thing this program needs to do is determine if an entered string is a **Palindrome**. To do this, you must complete the **isPal** method. For this version, the **isPal** method is *case-sensitive* meaning while *madam* and *MADAM* are palindromes, *Madam* is not.

You also are not concerned checking if the String is an *Almost Palindrome*. The program will generate **true** output for Almost Palindrome, but this can be ignored.

Two more examples of "Almost Palindromes" are shown below:

**A man, a plan, a canal, Panama Not A Banana Baton!**

**80 Point Version Output**



Note that all the outputs for "Almost Palindrome" are **true**.

This is normal and not an issue. If you only do the 80-point version, the only part that matters is

that your program can identify 100% true palindromes.

**100 Point Version Specifics**

The 100 point version is very similar to the 80 point version except now the **isPal** method is no longer *case sensitive*. So *madam* and *MADAM* are still palindromes, but now *Madam, mADAM*, and *mADam* are palindromes as well.

Al "Almost Palindrome" is a palindrome when the non-letter characters are removed.

For instance: "Madam, I'm Adam" is an "Almost Palindrome" when the non-letters are removed,

you will get the string "MadamImAdam" and after you change to uppercase it is "MADAMIMADAM",

which now checks out as a palindrome.

For this version you will need to complete methods **isLetter**, **purge** and **almostPal**. The conditions

of the three methods are explained in the student starting file.

**100 Point Version Output**

