/\*

\* Justin Mendes

\* Created: July 24, 2017

\* Last Edited: September 26, 2017

\* Unit 1 Activity 1 Program/Question 1

\* This program will display a rhyming triplet and prompt the user to add onto the possible lines

\*/

**import** java.util.ArrayList;

**import** java.util.Scanner;

**import** javax.swing.JOptionPane;

**public** **class** Triplets

{

**static** **int** *menu* = 1;

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

{

ArrayList<String> line = **new** ArrayList<String>();//has to be outside of the loop so it doesn't reset2

//for restarting program

**do**

{

//Variable Declarations and Initializations

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

**int** currentLine = 0, usedLine[] = **new** **int**[3];

String userAddedLine = "";

**boolean** store;

//populating the array with poetry lines

line.add("The goat raced to the hair");

line.add("So it could run the bear");

line.add("But it was a millionaire");

line.add("I didn't go there");

line.add("No more eating pears");

line.add("Jealousy makes notrils flare");

line.add("Too many shirts with rips and tears");

System.***out***.println("\nTriplet Maker\n============");

//reset or clear the usedLine variable

**for**(**int** i = 0; i < 3; i++)

{

store = **true**;

**do**//to make sure the same line does not pop up at the same time

{

**if** (store == **true**)//so usedLine gets the correct value ONLY once

{

usedLine[i] = currentLine;

}//end if

currentLine = *randomWholeNumber*(line.size());

store = **false**;

}//end loop

**while**(usedLine[0] == currentLine || usedLine[1] == currentLine || usedLine[2] == currentLine);

System.***out***.println(line.get(currentLine));//to display the poem using the accumulated 3 randomized lines

}//end loop

//to allow for repetitive additions to the poem lines

*menu* = Integer.*parseInt*(JOptionPane.*showInputDialog*(**null**, "Press 1 to see a new poem\nPress 2 to add a line to the poem maker\nPress anything else to exit", "Menu", JOptionPane.***QUESTION\_MESSAGE***));

**while**(*menu* == 2)

{

userAddedLine = JOptionPane.*showInputDialog*(**null**, "Type the line you would like to enter:", "New Poem Line", JOptionPane.***QUESTION\_MESSAGE***);

*menu* = Integer.*parseInt*(JOptionPane.*showInputDialog*(**null**, "Press 1 to see a new poem\nPress 2 to add a line to the poem maker\nPress anything else to exit", "Menu", JOptionPane.***QUESTION\_MESSAGE***));

line.add(userAddedLine);

}//end if

}//end loop

**while**(*menu* == 1);

}//end main

**public** **static** **int** randomWholeNumber(**int** highVal)

{

**return** (**int**) (Math.*random*() \* highVal);

}//end method randomWholeNumber

**public** **static** String userLine()

{

String newLine;

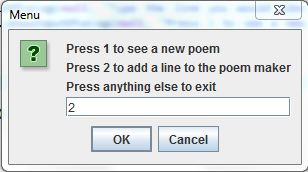
newLine = JOptionPane.*showInputDialog*(**null**, "Type the line you would like to enter:", "New Poem Line", JOptionPane.***QUESTION\_MESSAGE***);

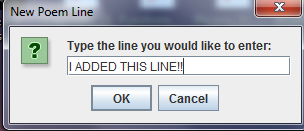
*menu* = Integer.*parseInt*(JOptionPane.*showInputDialog*(**null**, "Press 1 to see a new poem\nPress 2 to add a line to the poem maker\nPress anything else to exit", "Menu", JOptionPane.***QUESTION\_MESSAGE***));

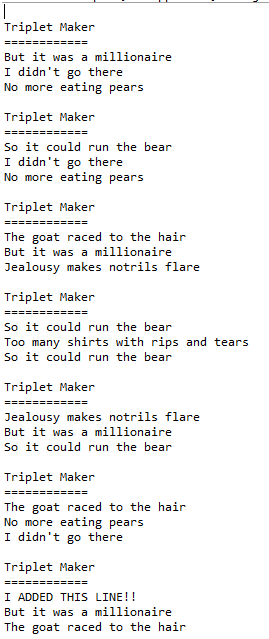
**return** newLine;

}//end userLine

}//end class







**import** java.util.Arrays;

/\*

\* Justin Mendes

\* Created: September 18, 2017

\* Last Edited: September 18, 2017

\* Unit 2 Activity 1 Program/Question 2

\* This program will print out the prime numbers from 1 - 1000 using The Sieve of Eratosthenes

\*/

**public** **class** Eratosthenes

{

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

{

//Variable Declarations and Initializations

Boolean numbers[] = **new** Boolean[1000];

**int** counter = 0;

//to initialize all booleans to true

Arrays.*fill*(numbers, Boolean.***TRUE***);

System.***out***.println("The prime numbers from 1 - 1000 are:");

**for**(**int** i = 2; i <= Math.*sqrt*(1000); i++)

{

**if**(numbers[i] == **true**)

{

**for**(**int** j = (**int**) Math.*pow*(i, 2); j < 1000; j += i)

{

numbers[j] = **false**;

}//end loop

}//end if

}//end loop

//Printing loop

**for**(**int** check = 2; check < 1000; check++)

{

**if**(numbers[check] == **true**)

{

counter++;

**if**(counter % 10 == 0)

{

System.***out***.println(check + " ");

}//end if

**else**

{

System.***out***.print(check + " ");

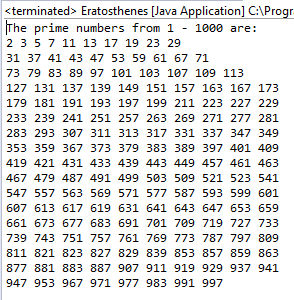
}//end else

}//end if

}//end loop

}//end main

}//end class



**import** javax.swing.JOptionPane;

/\*

\* Justin Mendes

\* Created: August 2, 2017

\* Last Edited: September 18, 2017

\* Unit 1 Activity 1 Program/Question 3

\* This program takes three colours, separated by hyphens, and prints the value of the resistor

\*/

**public** **class** Resistors

{

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

{

//Variable Declarations and Initializations

String userInput, colours[] = {"BLACK", "BROWN", "RED", "ORANGE", "YELLOW", "GREEN", "BLUE", "VIOLET", "GREY", "WHITE"}, userColours[];

**double** ohms[] = **new** **double**[3];

userInput = JOptionPane.*showInputDialog*(**null**,"What is your resistor's colour code?"

+ "\nSeparate each colour by hyphens\nOut of\n[Black, Brown, Red, Orange, Yellow, \nGreen, "

+ "Blue, Violet, Grey, or White]\nEx: Red-Orange-Black", "Resistor Colour", JOptionPane.***QUESTION\_MESSAGE***);

userColours = userInput.split("-");//to separate each word in the input and use them as needed

**for**(**int** i = 0; i < 10; i++)

{

**if**(userColours[0].equalsIgnoreCase(colours[i]))

{

ohms[0] += i \* 10;

}//end if

**if**(userColours[1].equalsIgnoreCase(colours[i]))

{

ohms[1] += i;

}//end if

**if**(userColours[2].equalsIgnoreCase(colours[i]))

{

ohms[2] = Math.*pow*(10, i);

}//end if

}//end for

System.***out***.println("Resistor Colour\n============");

System.***out***.println("You entered " + userInput);

System.***out***.println("The value of the resistor is " + (ohms[0] + ohms[1]) \* ohms[2] + " ohms.");

}//end main

}//end class

