Problem 1:

Use import declaration to prevent need for full qualification of Scanner class

Make a class called CalcAreaOfTriangle

Make a main method

Instantiate a new object of the Scanner class called scan

Declare all variables s, a, b, c, area, roundedArea as floats

Tell the user to input side a value

Use scan nextDouble method to record user input into variable a

Tell the user to input side b value

Use scan nextDouble method to record user input into variable b

Tell the user to input side c value

Use scan nextDouble method to record user input into variable c

Check to make sure all the sides can make a valid triangle

Perform computation for s

Calculate area using s

Use math round method to get a value with three decimal places

Print the roundedArea

Else print that the input side values do not make a valid triangle

A screen shot of a computer

Description automatically generated with medium confidence

Problem 2:

Use import declaration to prevent need for full qualification of Scanner class

Make a class called DetermineIfLeapYear

Make a main method

Instantiate a new object of the Scanner class called scan

Declare variables inputYear, remain4, remain100, remain400 as ints

Declare variables divis4, divis100, and divis400 as Boolean and initialize them to false

Tell user to input a year to check

Use scan nextInt method to record user input into variable inputYear

Determine remainder value when input year is divided by 4, 100, and 400

Check if input year is a year after Gregorian calendar is adopted

If it is, check if it is divisible by 4

If it is, set the bool divis4 to true

If it is a valid year, check if it is divisible by 100

If it is, set bool divis100 to true

If it is a valid year, check if it is divisible by 400

If it is, set bool divis400 to true

If divis4 is true check if it is divisible by 100 and not by 400 by checking the respective bools

If it is divisible by 4 and 100, but not 400, tell the user it is not a leap year

Otherwise, tell the user it is a leap year

Else it is not divisible by 4 (divis4 = false) tell the user it is not a leap year

And if the inputYear is less than 1582, tell the user this year was before the adoption of the Gregorian Calendar

A computer screen with white text

Description automatically generated with low confidence

Problem 3:

Make a class called CalcTimeToHalvePopulation

Create a main method

Declare current population constants with the final modifier as float data types

Declare the variables that will hold the population values as we increment the year (yearlyMexPop and yearlyUSAPop) as double data types.

Declare the variable t and initialize it to 0. t is an int data type and will represent years as we increment

Use a while loop that will iterate for as long as the Mexican population is less than half the US population

Within the loop, calculate the Mexican population with the math pow method using the equation P = P0(1+r/100)^t

Within the loop, calculate the US population with the math pow method using the equation P = P0(1+r/100)^t

Print that year’s population values for Mexico and US with the year from current date for reference

Increment the year (t++) for the next iteration of the loop

Once the Mexican population is greater than half the population of the US it will break the loop

Then print that it will require the current value in t minus 1 (due to the incrementation of t in the completion of the final iteration of the loop) for mexico’s population to reach half of the US’s.

A screen shot of a computer

Description automatically generated with low confidence