# Java String Exercises

## Before you start:

You have been given a skeleton project, containing a class called Week3Exercises. Implement a menu in the constructor of Week3Exercises which allows the user to choose which Task to run (call the corresponding method)

## Task 1:

Assign the string “A rose by any other name would smell as sweet” to a String variable called rosePoem.

Create an ArrayList<String> called individualWords and use the String split() method to get each word in rosePoem and add each word to the individualWords list.

Consider: what does the split() method return?

Consider: how does the split() method know where to split?

Print the words in individualWords to the console, one word per line.

Find and print the length of rosePoem – what does this length mean?

Change the rosePoem string to all upper case and print it out

Replace the word “rose” in the rosePoem with the word “daffodil”. What String method will you use for that?

## Task 2:

Ask the user to enter a some text, and store it in a variable called userString.

Tell the user:

* The string they entered
* How long the string they entered is
* How many words are in the string they entered (there are various ways of doing this, you could use split() or charAt() for instance)
* What character is third in the string they entered
* Whether or not the string they entered contained the word “carrot” (test this with and without carrot in the input string!)

Now find out whether any words are repeated in the string. Tell the user which words are duplicated, if any. This is a slightly trickier problem, there are a few things to think about and you might have to do some research:

* + How will you ensure your result is not skewed by case or punctuation?
  + How will you check for the duplicate words?
  + How will you store the words that are duplicated so you can tell the user about them?
  + Note: A HashMap might be useful in solving the last two questions

Create another string and assign “This is a very silly string, do better” to it.

Concatenate this string with userString (using the concat() method). Include a ‘\n’ character to make sure the resulting string displays across 2 lines.

Your output from this task may look something like this:

Text

Description automatically generated

## Task 3:

Use the String.format() method to format the following data into a string telling your user about John. Print the string.

Name: John

Age: 35

Monthly Wage; 1500.00

Number of children: 0

Now use the printf() method to print a similar string about Mary.

Name: Mary

Age 52

Monthly Wage: 450.00

Number of children: 3