

Part 4: Manipulating Strings

Solution Code

The trickiest bit about this code is doing the formatting to 2 decimal places of the final result. Once you have the result as a String, you can find the index of the decimal point, and then use this to get a substring which just has 2 more characters after the decimal point appears.

It's a bit subtle - and don't worry if you didn't get it! The point is that by stumbling across these types of requirements, you'll be able to add them to your toolbox of tricks you can use when you need to do some String manipulation in future.

Code Listings

App.java

```
package com.javaeasily.demos;

public class App {
    public static void main(String[] args) {
        System.out.println("Loan Calculator".toUpperCase());
        System.out.println();

        int amount = 100;
        int years = 5;
        double interestRate = 10;

        System.out.println("Calculating loan based on:");
        System.out.println("Initial Amount: " + amount);
        System.out.println("Number of Years: " + years);
        System.out.println("Interest Rate: " + interestRate + "%");

        double interestRateMultiplier = 1 + interestRate / 100;

        double year1AmountDue = amount * interestRateMultiplier;
        double year2AmountDue = year1AmountDue * interestRateMultiplier;
        double year3AmountDue = year2AmountDue * interestRateMultiplier;
        double year4AmountDue = year3AmountDue * interestRateMultiplier;
        double year5AmountDue = year4AmountDue * interestRateMultiplier;

        String totalAmountDue = Double.toString(year5AmountDue);
        int indexOfDecimalPoint = totalAmountDue.indexOf(".");
        String totalAmountDueFormatted = totalAmountDue.substring(0, indexOfDecimalPoint+3);

        System.out.println("Total Amount Due: " + totalAmountDueFormatted);
    }
}
```