

CET2041 - Backend Development with Java - Practicum 01

Topics Covered: Spring Core

Learning Objectives:

- Familiarize with the concept and implementation of Beans in Spring
- Familiarize with solving a problem with Spring

Deliverables:

- Submit a zipped Maven Project called `CET2041_P01_<Your_Name>.zip` (e.g. `CET2041_P01_John_Doe.zip`).
- Note that a non-working submission will result in a zero

Background

The Loans Department of Bank ABC would like to know which of their customers have defaulted on their loans. The loans can range from Car Loans, Home Loans, Study Loans, etc. A Loan is considered defaulted when the customer did not make a repayment in the last 30 days or more. These records are stored in `.csv` files with the following format:

```
1 | acct_number, balance, last_payment_date
```

Defaulted repayments has a negative balance.

so for something to be considered as defaulted, it needs to be negative and it needs to be

if in the csv file it is 01-01-2023 and today's date is 31-01-2023, it is considered as defaulted.

Task

Important notes

1. Make sure your Maven project is **cleaned** before zipping then submitting.
2. 2 CSV files will be provided: `carLoans.csv` & `homeLoans.csv`. You are advised to look through them before using them.
3. Test your completed program extensively.

You are to implement the program that accepts any number of CSV files of the same format detailed in the Background section and output the customer's details to terminal. Refer to *Test Cases* section. Do take note of the following requirements:

1. You have a choice between Spring `ResourceLoader` (you would have to carry out your own research) or `java.nio` libraries to use to load the CSV files into your program.
2. Your program has to be divided into several parts stored in different folders, each with an appropriate name to describe their contents.
3. Proper OOP concepts must be applied.

4. Program must not terminate in the event of an exception.
5. Your model object must be printable via the object itself.

```
1 CustomObj obj = new CustomObj();
2 System.out.println(obj);
```

this is talking about the pojo

Test Cases

The following are **non exhaustive** test cases to help test your program

Test Case 1

All records loaded okay.

Content in `homeLoans.csv`.

```
1 146-915,112.86,2019-09-22
2 322-338,-39.16,2020-11-25
3 807-807,-703.57,2019-01-25
4 922-574,-408.03,2021-11-07 <--- not defaulted yet
5 401-773,-745.99,2021-04-27
6 154-293,91.83,2020-06-21
```

Today's date: `28 Nov 2021`

Program output:

```
1 Defaulted for Home Loans -----
2 Acct: 322-338, Bal: -39.16, LastPaid: 2020-11-25
3 Acct: 807-807, Bal: -703.57, LastPaid: 2019-01-25
4 Acct: 401-773, Bal: -745.99, LastPaid: 2021-04-27
```

this should be Defaulted for homeLoans.csv as per below

Test Case 2

One of the CSV filename is misspelled.

Program output:

```
1 Defaulted for carLoan.csv -----
2 java.io.FileNotFoundException: class path resource [carLoan.csv] cannot be
  opened because it does not exist
3
4 Defaulted for homeLoans.csv -----
5 Acct: 322-338, Bal: -39.16, LastPaid: 2020-11-25
6 Acct: 807-807, Bal: -703.57, LastPaid: 2019-01-25
7 Acct: 401-773, Bal: -745.99, LastPaid: 2021-04-27
8 Acct: 720-261, Bal: -876.23, LastPaid: 2019-10-17
```

For the maximum allocation of marks, refer to the table below.

Description	Marks (%)
Successful implementation of the well tested program	50
Successfully load CSV files either using Spring <code>ResourceLoader</code> or <code>java.nio</code> libraries	20
Program has been be divided into several parts stored in different folders with proper names	10
Applied OOP concepts	5
Model object printable via the object itself.	5
Proper and consistent naming conventions.	5
Proper and sufficient comments to explain code.	5