Java CLD - Internship/Summer Practice - July-August 2022

Calendar

No.	Chapter	Training Title	Problem Title	Scheduled	Date w/c
1	Java				
	Intro to Java	Modern Java: The Big Picture by Sander Mak	Chapter 1: Design your coffee	Week 1	18 Jul 2022
	Fundamentals	Java Fundamentals: The Java Language by Jim Wilson	Chapter 1: Design your coffee	Week 1	18 Jul 2022
2	Spring				
	Basics	Spring Fundamentals by Bryan Hansen	Chapter 2: Inject me some expresso	Week 2	25 Jul 2022
	Spring Boot	Spring Boot Fundamentals by Kesha Williams	Chapter 2: Inject me some expresso	Week 2	25 Jul 2022
	Dependency management (maven)	Maven Fundamentals by Bryan Hansen	Chapter 2: Inject me some expresso	Week 3	25 Jul 2022
3	REST	REST Fundamentals by Howard Dierking	Chapter 3: Let us pay via card & Secure my hot coffee	Week 3	01 Aug 2022
4	Database				
	DB: Relational	Relational Database Design by Hugo Kornelis	Chapter 4: Save my preferred cup of java	Week 4	08 Aug 2022

Participants

No	Participant	Email	University	Total (h)	Expert buddy	Start date	End date
1	Bianca Zelenszky	mailto:Zelenszky.Sa.Bianca@student.utcluj.ro	UTCN	200	Nicolae Giuroiu	11 Jul 2022	26 Aug 2022
2	Theodor Gheorghe Marian	Marian.Ni.Gheorghe@student.utcluj.ro	UTCN	200	Adrian Mihai Pop	11 Jul 2022	26 Aug 2022
3	Crisan Cristian Sidon	crisan.al.cristian@student.utcluj.ro	UTCN	200	Denisa Melisa Radu	11 Jul 2022	26 Aug 2022
4	Pop Emanuel Bogdan	Pop.Da.Emanuel@student.utcluj.ro	UTCN	240	Bogdan Iudean	11 Jul 2022	02 Sep 2022
5	Baciu Norbert Sorin	Baciu.So.Norbert@student.utcluj.ro	UTCN	240	Jozsef Endre Kerekes	11 Jul 2022	02 Sep 2022
6	Ana-Maria Ppar	ana.papara@stud.ubbcluj.ro	UBB	120	Victor Gheorghe Georgescu	11 Jul 2022	05 Aug 2022
7	Milan Racz	Racz.Ti.Milan@student.utcluj.ro	UTCN	200	Vlad Ioan Ovadiuc	11 Jul 2022	26 Aug 2022
8	Corbean Mircea	Corbean.Te.Mircea@student.utcluj.ro	UTCN	240	Alexandru Branea	11 Jul 2022	02 Sep 2022
9	Iulian Bogdan	iulian.bogdan@stud.ubbcluj.ro	UBB	120	Sergiu Prodan	11 Jul 2022	05 Aug 2022

Program coordinator: Dariana Lupea

Curriculum

The curriculum used can be found here: Java CLD - Internships - Curriculum

Problem statement

A fresh new coffee shop would like to offer its wonderful beverages to the world.
Its customers will need a way to pay by card. Whether it's a Latte or a Triple Mocachinno, a pick-up or delivery, all customers should pay in a modern way.

Chapter 1:
Design your coffee

User Story#1: As a new coffee shop, we would like to have Customers so that we can share our recipes with the world.
Acceptance Criteria:

The most important citizen in our world is the Customer.

Any good coffee starts with a coffee base and optionally more ingredients like milk, syrup, honey.

Specific details of contents can be found here:
https://thumbs.dreamstime.com/z/recipes-most-popular-types-coffee-their-preparation-flat-style-vector-icon-set-illustration-infographic-69329512.jpg

The coffee shop needs a list of 5 beverages: Espresso, Machiatto, Coffee Latte, Cappuccino, and Coffee Miel.
Each of these beverages has a specific recipe for their content. For example: Espresso has 1 expresso shot, while Machiatto adds Milk Foam on top.
Each coffee has a price and the shop is keen on summing up their profit.

Coffee orders may be a pick-up (to be consumed in the shop) or delivery (to-go).
Each ordered beverage should be identifiable by their customer's name.

Print the coffee shop name and list of beverages.
Take user input for selecting a specific coffee.
For each order, print the price of the coffee, the ordered beverage, and the shop's profit for the day.

User Story#2: As Customers, we would love to design our coffee so that we can start better our day.
Acceptance Criteria:
Any customer may choose to design their own coffee. They will be prompted about the number of coffee shots and any additional ingredients.
The price of the beverage will be a sum of the prices of all ingredients.
And the beverage will have the name of its owner.
For each order, print the price of the coffee, the ordered beverage, and the shop's profit for the day.

Chapter 2:
Inject me some expresso

User Story#3: As a new coffee shop, we would like to service our Customers to brew their own coffee.
Acceptance Criteria:
Your application will have a specialized service for brewing coffee.
As input, it will take a coffee recipe that contains specific amounts for each ingredient.
The output will be a fresh coffee.

User Story#4: As a new coffee shop, we would like to validate our stock so that we don't run out of supplies.
Acceptance Criteria:
The shop will utilize a service that checks the stock of supplies. All ingredients for making a coffee are considered supplies of the shop, together with sweeteners.
On a scheduled basis, the shop wants to run a stock inventory to check whether it runs low on some ingredients.
Print the quantity of each ingredient and a warning if there is no more supply for the next 3 coffees.

Chapter 3:
Let us pay via card & Secure my hot coffee

User Story#5: As Customers of the coffee shop, we would like to pay by card our orders so that we don't need to carry any cash.
Acceptance Criteria:
Implement a RESTful API for ordering a coffee. The customer will be paying via card.
The new endpoint for this will be:
POST /orders/pay
and it will take as input a card number, cardholder name, expiry, and CIV, together with order details.
Order details are beverage name, quantity, customer name.

User Story#6: As the Coffee shop, we need to receive valid card data so that we prevent fraud.
Acceptance Criteria:
A new service will permit the validation of the card data using the Luhn algorithm.
If the card data is not valid, the order will not go through and the error will be printed.

Chapter 4:
Save my preferred cup of java

User Story#7: As the Coffee Shop, we need to keep an inventory of all the ingredients we have on stock so that we know when we run out of them.
Acceptance Criteria:
Keep the inventory in a database.
Each time a new order is executed, the stock decreases correspondingly. We imagine that you would populate your database with this data via a script and use Read operations for working with the elements.

User Story#8: As the Coffee Shop, we care about our precious recipes and we want to store them so that we don't lose any of them.
Acceptance Criteria:
Store the coffee recipes in the database. We imagine that you would populate your database with this data via a script and use Read operations for working with the elements.

User Story#9: As the Coffee Shop, we need our orders saved for audit reasons.
Acceptance Criteria:
Save each order with information about customer name, date and time, order type: pick-up/to-go, order content: beverage name, quantity, price and total order amount.

Guidelines

Ways of working

Each participant will receive a private git repository, identified by the URL in the section above.

The programme's expectation is that each week, participants will learn the theoretical topic by following the **Learning Path** and they will implement the corresponding chapter from the **Problem statement.**

Each problem chapter is mapped to the corresponding week's theoretical concepts, in order to provide students a practical learning experience.

Each chapter of the Problem statement has a set of User Stories that need to be implemented on a separate git branch, branched off the develop branch.

This means that there will be a separate branch for each User Story.

When the work is done on a user story branch, a new Pull Request will be created and sent to the expert buddy for review.

After implementing the code review changes, the user story branch will be merged into the develop branch.

All work will be presented at the end of the programme, in a demo session.

Naming conventions

The following naming convention will be used:

```
US#N-SHORT-NAME
```

where:

- "US#" is a constant string followed by the user story number N and "-" symbol

- SHORT-NAME is a short string denoting the user story requirement, multiple words will be separated by the "-" symbol e.g.: US#12-TEST-EVERYTHING

Also, for commit messages, the following naming convention will be used:

```
[US#1] First draft of shop
```

Rules

All participants must respect the naming conventions, failure to do so will incur negative feedback.

All work needs to be submitted on time, for each corresponding week. Failure to present the entire work in the demo will lead to negative feedback.

Please note that any discovered attempts at cheating/copying will lead to instant disqualification from this programme.