

# INF1009 Object-Oriented Programming – Lab 01

## 1. Introduction

Welcome to your first lab sheet for INF1009 Object Oriented Programming! In this lab we will establish all the basics needed to get you started in this module. This lab acts as a foundation for all the subsequent labs, so it is very important to complete it but even more important to understand everything presented here. In this lab, you will create a simple Retail Management System using Java. The application will allow a store manager to add items, calculate total inventory value, apply discounts, and generate reports. This integrated exercise combines all foundational OOP concepts. If, you have any doubts about anything, do ask your prof for some support.

Remember, the focus of this module is all about object-oriented programming and good programming practices. We will evolve our code from week to week based on the topics covered in class each week. You will be required to submit your code every week as labs will be graded, and the expectation is that you complete the lab by the end of the same day as the lab session.

Further, learning and understanding is most important, so do ensure you work through and complete each lab by yourself, peers may help with your understanding but shouldn't write code for you or provide code to you!

It is very important to ensure you have watched the weekly lessons up until this point before attempting this lab as we will assume you have gained knowledge from those sessions which you can now use to understand this lab's activities.

Remember to ask plenty of questions to your prof and teaching assistants when you get stuck, we are here to help!

## 2. Learning outcomes

The following lab session are designed to familiarize with following topics:

- a) Run and structure Java console applications.
- b) Use variables, primitive types, Strings, and formatted console I/O.
- c) Apply decision logic (if/else, switch) and loops (for, while) to implement rules.
- d) Design and call static methods; compose functions and validate inputs.

### 3. Lab Assignment

#### 3.1 Project Setup

Begin by creating a new Java project and adding a main class named `RetailApp.java`. Inside this class, define the public static void `main(String[] args)` method as the entry point for your application. Import the necessary packages such as `java.util.Scanner` for user input and `java.util.ArrayList` for managing collections of items. This setup ensures you have the basic structure and tools ready to build the retail management system.

#### 3.2 Capture Input

Ask the user to enter the store name and starting budget. Use the `Scanner` class to read these values from the console. After getting the inputs, show a welcome message with the store name and budget, for example:

*Welcome! Current budget: \$5000.00.*

This step teaches how to read input and display formatted output.

#### 3.3 Implement Menu

Design a menu-driven interface that allows the user to perform different operations repeatedly until they choose to exit. The menu should include options such as:

1. Add Item
2. View Inventory
3. Exit

Use a while loop to keep displaying the menu and a switch statement to handle the user's choice. This step demonstrates control flow and looping constructs.

#### 3.4 Add items

Implement functionality to add new items to the inventory. Prompt the user for SKU, name, price, and quantity. Store these details in separate `ArrayLists` (one for each property) or use a single `ArrayList` of strings for simplicity. Validate that price and quantity are positive.

#### 3.5 View Inventory

Display all items in a table-like format using the data stored in your lists. Show columns for SKU, name, price, and quantity. At the end, calculate and display the total inventory value by looping through the price and quantity lists. This step teaches iteration and formatted printing.

#### 3.6 Exit

When the user selects "Exit," display a goodbye message and stop the loop. Close the `Scanner` before ending the program. This step ensures clean program termination.

## 4. Sample Output

```

Enter store name: Zara
Enter starting budget (>= 0): 100000
Welcome to Zara! Current budget: $100000.00

===== Retail Management System =====
1. Add Item
2. View Inventory
3. Exit
Choose an option (1-3): 2
--- Inventory ---
No items in inventory.

===== Retail Management System =====
1. Add Item
2. View Inventory
3. Exit
Choose an option (1-3): 1
--- Add Item ---
Enter SKU: 345
Enter item name: Jeans
Enter price (> 0): 100
Enter quantity (> 0): 50
Item added successfully.

===== Retail Management System =====
1. Add Item
2. View Inventory
3. Exit
Choose an option (1-3): 1
--- Add Item ---
Enter SKU: 321
Enter item name: Shirt
Enter price (> 0): 50
Enter quantity (> 0): 100
Item added successfully.

===== Retail Management System =====
1. Add Item
2. View Inventory
3. Exit
Choose an option (1-3): 2
--- Inventory ---

```

SKU	Name	Price(\$)	Qty	Line Total
345	Jeans	100.00	50	5000.00
321	Shirt	50.00	100	5000.00

```

-----
Total Inventory Value: $10000.00

===== Retail Management System =====
1. Add Item
2. View Inventory
3. Exit
Choose an option (1-3): 3
Goodbye! Exiting program.

```

## 5. Submission

Once you have completed all the activities in this lab, you may zip your root project folder which will contain all of your project setup files and code files. This can then be submitted to the dropbox for **Lab01** which you will find on XSITE.

Remember, we expect you to complete the lab by the end of the day on which your lab takes place.