Final Project

Estimated Duration: 30 minutes

Learning Objectives

After completing this project, you will be able to:

- Understand how to create Java program
- Handle Strings and string operations
- Work with Operators and different Data Types
- Handle Exceptions and throw new excpetions
- Create `for` Loops and `while` Loop
- Check with Conditional statements
- Iterate through Arrays
- Create basic methods and functions

About the course project

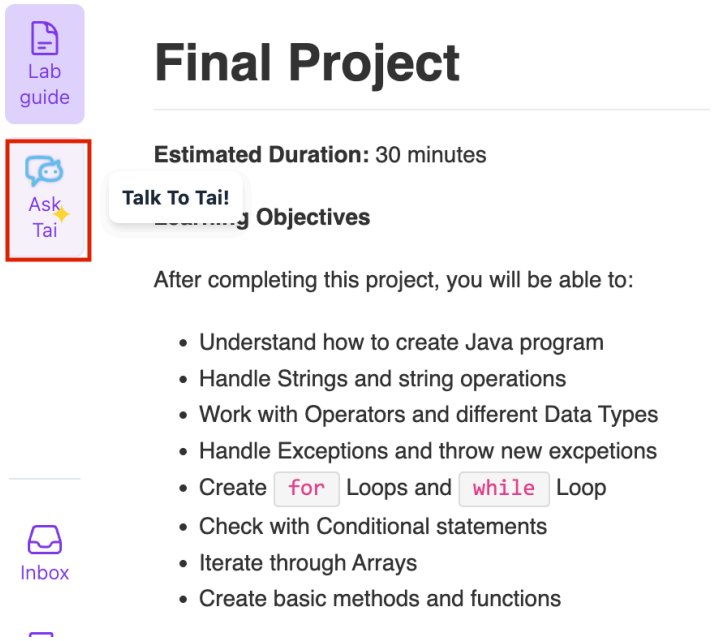
Welcome to the final project for your first Java Development course.

In this project, you will apply the knowledge and skills learned in this course to a simulated scenario.

The tasks in this hands-on project correspond to the activities performed by a Java Developer who is creating a stand-alone console application.

In this project, you will create a console application for a Grocery shop, which will calculate the totals for grocery items chosen, depending on the unit price and quantity. You will receive guidance to complete this part of the project.

To further challenge your learning, the application you develop will also allow searches by name, filtering by price, calculate average price of an item in the list, provide discounted billing for customers who meet a specific condition and provide inventory management. This lab provides instructions for these tasks. You are encouraged to **Ask Tai**, your AI teaching assistant for help completing the challenge section.



You are currently viewing this lab in a Cloud based Integrated Development Environment (Cloud IDE). It is a fully-online integrated development environment that is pre-installed with JDK 21, allowing you to code, develop, and learn in one location.

GroceryShopping application

1. Create a project directory by running the following command.

```
mkdir final_proj
```

2. Run the following code to create the directory structure.

```
mkdir -p final_proj/src
mkdir -p final_proj/classes
mkdir -p final_proj/test
cd final_proj
```

3. Now create a file named `GroceryShopping.java` inside the src directory.

```
touch /home/project/final_proj/src/GroceryShopping.java
```

4. Select the following button to open the file for editing.

Open **GroceryShopping.java** in IDE

Start adding the code

1. Create a public class with a public main method.

▶ Click here for sample code

2. Create a String array of items you will buy from a grocery shop. It should be a minimum of 10, but not more than 25.

3. Create a float array of unit price for items, which corresponds to the index positions in the items array. It should be the same length as the items array.

▶ Click here for sample code

4. Import and create a scanner object and create an object of the scanner obect to read from the console.

▶ Click here for sample code

5. Create an infinite loop which runs as long as the user wants to. The loop should exit when the user inputs `Exit` (ignoring the case).

▶ Clcik here for sample code

6. Create an infinite loop which runs as long as the user wants to add items. The loop should exit when the user inputs `Finish` (ignoring the case) else check the array for the item. If the item is in the array, get the index of the item and store it. If not, throw an exception.

▶ Click here for sample code

7. Retrieve the item price from the same index position in the array as the item. Calculate the price for each item and add each price to the total bill.

▶ Click here for sample code

8. After chosing all of the required items, print the total bill price.

Sample code

You can implement this solution using many methods. The code provided here is for you to use as a structure, based on the tasks you need to complete thus far.

▶ Click here t view sample application code for this stage

Additional challenges

In this part of the project you are expected to complete the following tasks:

- Implement item search functionality
- Calculate average price
- Filter items below a certain price
- Calculate the total bill with discounts
- Implement inventory management

Item Search Functionality

1. Write a method called `searchItem` that takes a String array of items and a String item name as parameters.
2. Use a for loop to iterate through the items array.
3. If the item is found, print its index position. If the item is not found, print “Item not found.”
4. Call this method within your main program and test the method using different item names.

Calculate Average Price

1. Write a method called `calculateAveragePrice` that takes a float array of prices as a parameter.
2. Use a for loop to sum all the prices in the array.

3. Divide the total of all prices by the length of the array to find the average.
4. Return the average price and print the average price in your main program.

Filter Items Below a Certain Price

1. Write a method called `filterItemsBelowPrice` that takes a String array of items and a float array of prices, along with a float threshold price.
2. Use a for loop to check each price against the threshold.
3. If the price is below the threshold, print the corresponding item name.
4. Call this method using different threshold prices in your main program.

Total Bill with Discounts

Use conditional statements to implement the following logic:

1. After calculating the total bill, check if the total exceeds $100.
2. If the total exeeeds $100, apply a 10% discount on the total bill.
3. Print both the original total and the discounted total.

Inventory Management

Implement the following logic within your purchase loop.

1. Create an integer array named **stock** that corresponds to your items array, representing the stock available for each item.
2. After a purchase is made, decrease the stock for that item by the quantity purchased.
3. If a user tries to purchase an item that has insufficient stock, print a message indicating that the item is out of stock.

Author(s)

Lavanya