**50002870006 - Java Enterprise Application Development**
**50002870008 - Java Enterprise Application Development (Hons)**
**Academic Year 2024-2025 Semester 1**

---

# Programming Examination

**Time Slot: 1:30-4:15pm, 30 October 2024**
**Venue: Lab 430, Jishi Building, Tongji University**

**Notice:**

1. Compress all source code folders and files into one ZIP file, and submit it via the *Canvas* system. No binary file (such as .class/.jar file) or IDE-related file should be submitted.
2. When it reaches 4:15pm, you must immediately stop your programming work and initiate the submission process. You will be given 15 minutes for uploading your code and solving technical issues encountered. The *Canvas* submission portal will be automatically closed at 4:30pm.
3. When solving the problems, please also pay attention to the design, readability and maintainability of your source code.
4. Please write your matriculation number and full name (as source code comments) in the beginning of each source code file submitted.
5. You may read technical documents and search for information over the Internet during the examination. However, it is STRICTLY NOT ALLOWED to use any communication service (such as sending/receiving emails, asking for help in a forum, using WeChat/QQ), and STRICTLY NOT ALLOWED to use any AIGC service (such as ChatGPT) regardless of the form.
6. During grading, JDK 21 will be used for testing your code.

# Garfield Restaurant

**(100 Marks)**

Garfield enjoys eating pizza and often visits a restaurant that serves a variety of food such as pizza, fried chicken, and french fries, with diverse flavors, features and prices. He needs a comprehensive system to view the food information and place orders.

Now, your task is to assist Garfield to implement a restaurant information system with object-oriented programming principles and skills. When your program is launched, it firstly extracts food information from an external data source (by utilizing specified Java libraries), and properly store them in memory (as described by Task 1 in details). Consequently, your program provides a command-line based simple user interface for supporting functionalities on food information display, food ordering, and PDF report generation (as described by Tasks 2, 3 and 4 in details).

## Detailed Tasks

Design and implement the *Garfield Restaurant*, with the following functionalities and features:

### 1. Data Retrieval and Storage (40 Marks)

- **Data Retrieval**: Utilize the *jsoup* library (see reference) to extract details of various types of food (e.g. pizza, fried chicken, french fries) from the given *food.html* file. Retrieve food attributes such as *name*, *weight*, *price*, *features*, *calories*, and *image*. Please note that, in addition to the above common attributes, each food type may have its unique attributes (such as radius for pizza, spiciness for fried chicken, and thickness for french fries). In addition, you should consider potential extensions for supporting other data sources in the future, such as a Web API or a JSON/XML plain text file.
- **Data Storage**: Design and implement appropriate data structures to store information of each food item. Particularly, for the image of each food item, you may simply store the link (as a string) rather than the image data. Please also consider potential extensions on more food types and items in the future.

### 2. Food Information Display (20 Marks)

- **Food List Display**: Design and implement a command-line based interface with the functionality to display lists of different food types. Each list should demonstrate all attributes of its food type. Besides, the interface also allows end-users to select which food type to view.
- **Sorting Support**: Provide flexible sorting options for food list display. The program should support sorting by *calories* and *prices* respectively, and be implemented with consideration of potential extensions on more sorting criteria in the future.

**3. Food Ordering with Discount Plan (20 Marks)**

- **Food Ordering**: Allow the end-user to order food items one by one, and eventually print the order information, including the total price and calories.
- **Applying Discount Plan**: When the order is being placed, allow the end-user to select and apply one discount plan. By default, the restaurant provides two built-in discount plans, including (a) deducting $50 off when the sum is above $100, and (b) applying 50% off for Margherita Pizza only. Please also consider potential extensions on more discount plans in the future.

**4. PDF Report Generation (20 Marks)**

- **PDF Report Generation**: Provide a functionality for generating a PDF report that describes the information of a placed order, which contains: (a) the food items contained (including the quantity, price, and calories of each item), (b) the discount plan applied, (c) the total price and calories of the order, and (d) detailed descriptions (i.e. attributes) of each food item in the order.
- **PDF Generator Integration**: The above PDF report generation is technically implemented by utilizing third-party PDF libraries. By default, your program is integrated with two libraries, namely *iText Core* and *Apache PDFBox* (see references), and the end-user is allowed to choose one of them for report generation. In addition, please also consider possible extensions for supporting more PDF libraries in the future.

## Important Notice:

- **Design consideration**: The implementation of your program should follow good object-oriented design principles and considerations, such as well-designed class hierarchies, careful design on class encapsulation (such as proper design of methods/fields and constructors), the use of polymorphism, and the idea of code reuse.
- **Error handling**: Implement error handling to manage issues such as missing data and file I/O errors.
- **Code documentation**: Provide proper documentations for your code, including source code comments and a README file explaining how to use the program from an end-user's perspective.

## References


- jsoup: https://jsoup.org
- iText Core: https://itextpdf.com/products/itext-core
- Apache PDFBox: https://pdfbox.apache.org


*The End*