**COMP1202 - Programming 1**

**Coursework 2**

**Name: BRANDON TING WEE KANG**

**Student ID: 33399085**

**Choice: B. Food Ordering System**

**Part 1. Relations Tree**

**Draw a simple tree that shows the relation between the classes. The diagram must show the class type (abstract or concrete), and the relationship type (implements, extends, or composed of).**

Answer:

To create a relations tree for the given Java code, consider the following structure:

1. **MenuItem (Abstract Class)**: Extended by **FoodItem** and **DrinkItem**.
2. **Order (Concrete Class)**: Composed of **User**, **Restaurant**, **CartItem**, and **SpecialOffer**.
3. **Cart (Concrete Class)**: Composed of **CartItem**.
4. **CartItem (Concrete Class)**: Contains **MenuItem**.
5. **Restaurant (Concrete Class)**: Composed of **MenuItem**, **SpecialOffer**, and **Inventory**.
6. **Inventory (Concrete Class)**: Contains **MenuItem**.
7. **SpecialOffer (Concrete Class)**: Used in **Order**.
8. **User (Concrete Class)**: Contains **Order**.

The relationships are as follows:

1. **MenuItem** is an abstract class.
2. **FoodItem** and **DrinkItem** extend **MenuItem**, demonstrating inheritance.
3. **Order**, **Cart**, **CartItem**, **Restaurant**, **Inventory**, **SpecialOffer**, and **User** are concrete classes.
4. Composition is shown in classes like **Order**, which is composed of **User**, **Restaurant**, **CartItem**, and **SpecialOffer**.

**Part 4. Storytelling**

**1. What problems did you encounter during the development of this coursework; what makes you stuck for many hours trying to figure out what's wrong.**

*During the development of this coursework, significant challenges arose in handling user input, testing and debugging, and data management. Interacting with users in a command line interface posed complexities, requiring meticulous attention to diverse input options and robust error handling. Testing and debugging proved time-consuming, particularly when troubleshooting interactions between various components. Efficient data management, encompassing inventory and order history, demanded careful consideration of data structures and storage mechanisms. These challenges collectively posed hurdles that required prolonged efforts to navigate and find optimal solutions, contributing to a comprehensive learning experience in software development.*

**2. How did you manage to solve the problems.**

*Addressing these challenges involved implementing systematic solutions. To manage user input challenges, I established robust input validation and error-handling mechanisms, employing loops and conditionals to guide users and ensure valid inputs. Debugging complexities were mitigated by leveraging debugging tools to step through the code and identify issues, complemented by the creation of unit tests for functions and components, facilitating early error detection. For efficient data management, I opted for appropriate data structures like lists and maps and considered data persistence options, tailoring choices to the project's needs, while prioritizing data consistency and integrity. These solutions collectively streamlined the development process and contributed to the overall success of the project.*

**3. What are the things that you are proud of about your program; imagine this is a business pitching session.**

*I am proud of the data management capabilities of my program. The administrator parts have seamless control, with the ability to effortlessly add, delete, modify, and view restaurant and menu data. This feature empowers administrators to efficiently curate the platform's offerings. Furthermore, customers experience a smooth and user-friendly interface, coupled with the convenience of viewing their comprehensive order history. This dual functionality not only enhances the overall user experience but also underscores the program's excellence in data management, positioning it as a reliable and versatile solution for both administrators and customers alike.*