Department of Computer Science, Mathematics & Physics

COMP2232 – Object-oriented Programming Concepts
Semester 2, 2020-21: Assignment #2

Read this entire document before starting your assignment!

This assignment is a group assignment and you are expected to submit your group's own work.

Any indication of **any form of copying or plagiarism will be dealt with severely**.

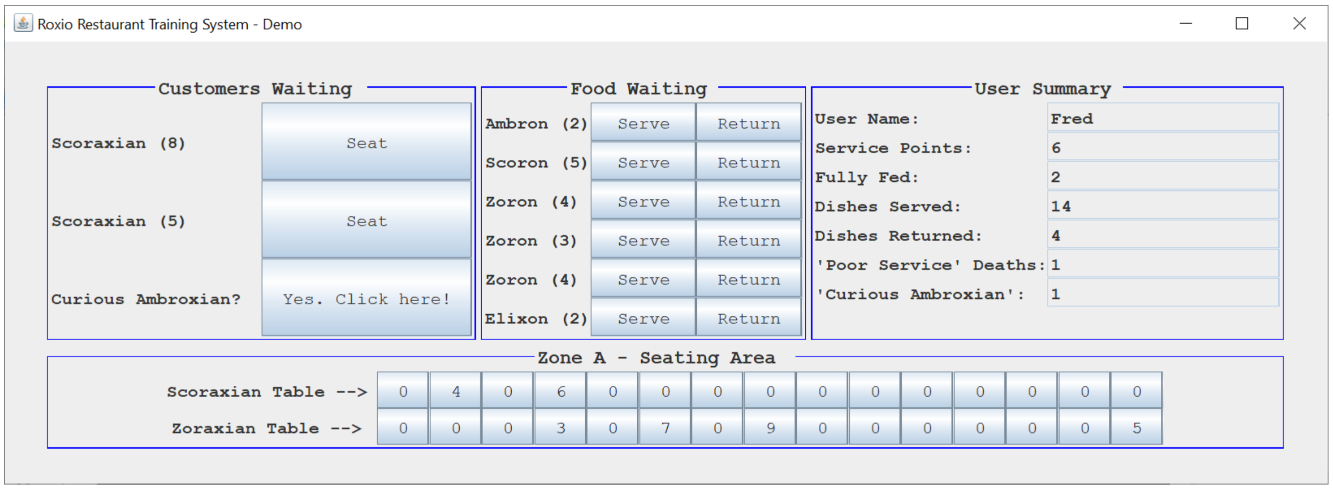
Roxia Restaurants

The purpose of this assignment is to provide further exposure to object-oriented concepts, the Java language and the implementation of graphical user interfaces.

The owners of Roxia Restaurants would like to continue to work with you on implementing the training program.

You will be required to perform file processing, allow system play (seating & feeding customers, scoring, etc…) and the production of a user performance report. This will involve a commitment to learning and some research.

The main project interface will be comprised of 4 sections (panels): Customers Waiting, Food Waiting, User Summary, Zone A – Seating Area. (*see image below*)



Roxio Restaurant Training System - Demo

Customers Waiting

Scoraxian (8)	Seat
Scoraxian (5)	Seat
Curious Ambroxian?	Yes. Click here!

Food Waiting

Ambron (2)	Serve	Return
Scoron (5)	Serve	Return
Zoron (4)	Serve	Return
Zoron (3)	Serve	Return
Zoron (4)	Serve	Return
Elixon (2)	Serve	Return

User Summary

User Name:	Fred
Service Points:	6
Fully Fed:	2
Dishes Served:	14
Dishes Returned:	4
'Poor Service' Deaths:	1
'Curious Ambroxian':	1

Zone A - Seating Area

Scoraxian Table -->	0	4	0	6	0	0	0	0	0	0	0	0	0	0	0
Zoraxian Table -->	0	0	0	3	0	7	0	9	0	0	0	0	0	0	5

The following will explain how this version of the training is expected to work and the functionality of each panel.

The Restaurant – Zone A

- caters to Zoraxians and Scoraxians only. Ambroxians are advised not to enter (however, sometimes they ignore the signs and wander in like curious cats – it ends the same way for them as it does for the cats).
- there are two long tables for seating: one table for Zoraxians and the other for Scoraxians. Each table has **15 seats**. More seats may be added to this restaurant at some time in the future.
- **Only** one diner may occupy a seat at any point in time.

- The restaurant serves **zoron**, **scoron**, **ambron** & **elixon**.

Scoraxians

- They have a maximum energy level of 20. Exceeding this results in 'death by greed'.
- They are predators who find weak Zoraxians easy pickings.
- They eat scoron and ambron. They are highly allergic to zoron (consuming it results in 'death by allergic reaction'). Consumption of any amount of elixon depletes their energy level by 25%.

Zoraxians

- They have a maximum energy level of 15. Exceeding this results in 'death by greed'.
- They become weak and defenceless if their energy level falls below 3.
- They eat zoron and ambron; can also eat scoron but cannot digest it fully resulting in gaining only half of its energy value. Consumption of any amount of elixon increases their energy level by 25%.
- They are natural hunters and consider Ambroxian's a suitable meal.

How the program works

Setting up the System

- To begin, the user will be required to enter their name.
- The tables in the *Seating Area* Zone A panel are randomly populated with 8 diners (any number of Zoraxians and Scoraxians totalling 8.)
    - **Refer to the diagram on page 1:** each button is a table. The value on the button represents the energy level of the customer sitting there. Energy levels of 0 represent an empty seat.
    - Scoraxians are initialized with a random energy level between 5 and 10 inclusive.
    - Zoraxians are initialized with a random energy level between 4 and 7 inclusive.
- The *User Summary* panel is initialized.
    - **Refer to the diagram on page 1:** Service Points, Fully Fed, Deaths, etc are initialized to 0 and displayed on-screen with the user's name.
- The *Customers Waiting* panel will not have any customers waiting.
    - **Refer to the diagram on page 1:** These fields will contain "Empty Queue" instead of a Customer Type; the *Seat* buttons will be disabled; the Ambroxian button will be labelled "No" and disabled.
- The **Food Waiting** panel will display the first 6 dishes which the chef has prepared (1 ambron, 1 scoron, 1 zoron, 3 random), each with a random energy value within a range of 2 to 5 inclusive.
    - **Refer to the diagram on page 1:** The list of 6 dishes (type with energy in parentheses) will be displayed. (**Recall:** restaurant serves ambron, elixon, scoron & zoron).
    - The *Serve* and *Return* buttons next to each dish will be enabled.

The Play

- There will be 7 rounds.
- **To start the first round**, the user will need to use the food which the chef has prepared. For each dish, they may either choose to *Serve* the dish or *Return* it to the kitchen.

- If the *Return* button next to a dish is selected, that dish is discarded and both the *Return* and *Serve* buttons are disabled for that dish. The number of *Dishes Returned* will be updated in the *User Summary* panel.
- If the *Serve* button next to a dish is selected, the user will then have to select a customer to feed the dish to in the *Seating Area*. This is done by selecting the button representing the customer to be fed. Once this is done, both the *Return* and *Serve* buttons are disabled for that dish. The number of *Dishes Served* will be updated in the *User Summary* panel. The energy level for the customer will be updated appropriately and displayed on their table button.
- Any diners who have overeaten (exceed energy level) will throw a GreedyGutsException and be removed from the table (update *Seating Area*). 3 Service Points will be deducted from the user (update *User Summary*). An appropriate "Poor Service" message will popup indicating incident details (customer type, seat number, overeating). An **incident record** is also stored to be added to the final performance report.
- Any diner who has been fully fed (reached max energy level) will be removed from the table (update *Seating Area*). 2 Service Points will be awarded to the user (update *User Summary*). An appropriate "Good Service" message will popup indicating details (customer type & seat number, fully fed).
- Any weak Zoraxians (energy < 3) will be consumed by the Scoraxian seated opposite. If this seat is empty the nearest Scoraxian (to the right) will consume it. 3 Service Points will be deducted from the user (update *User Summary*). An appropriate "Poor Service" message will popup indicating incident details (customer type, seat number, weak eaten by Scoraxian, Scoraxian seat number). An **incident record** is also stored to be added to the final performance report
- Any Scoraxians who were served zoron will throw an AllergyException and be removed from the table (update *Seating Area*). 3 Service Points will be deducted from the user (update *User Summary*). An appropriate "Poor Service" message will popup indicating incident details (customer type, seat number, what caused death). An **incident record** is also stored to be added to the final performance report
- The first round ends when all dishes have either been served or returned to the kitchen.

- **For the subsequent rounds:**
  - The system will randomly add 2 more diners (Zoraxian and/or Scoraxian) to the *Customers Waiting* panel, forming a queue. The *Seat* buttons next to them will be enabled.
  - The system will randomly determine if an Ambroxian will wander into the restaurant or not.
    - If one does not, the *Customers Waiting* panel will display "No" on the button next to the label *Curious Ambroxian?*; the button will be disabled.
    - If one does, the button will display "Yes. Click here!" and the button will be enabled.
  - The chef will prepare another combination of 6 dishes, as before. (1 ambron, 1 scoron, 1 zoron, 3 random). These will be displayed and the Serve and Return buttons enabled.
  - The user may proceed to seat and feed customers. This may be done in any order.
  - **Seating customers:** click on the *Seat* button next to the customer in the *Customers Waiting* panel and click on the desired seat in the *Seating Area* panel. That customer's *Seat* button is disabled once the customer is successfully seated.
  - Clicking on the "Yes. Click here!" in the *Customers Waiting* will allow the diner with the lowest energy to eat the Ambroxian resulting in a 2-point energy increase for the lucky diner. An appropriate "Curious Ambroxian" message will popup indicating incident details (seat number & type of customer who ate the Ambroxian).
  - **Feeding customers:** same as round one.
    - Any diners who have overeaten, are fully fed, become weak or have had an allergic reaction will be handled as in round one.

- **At the end of every round**, any diner in the Customers Waiting queue will become hungrier with every round they are still in the queue (deplete energy by 1). Any with an energy level < 4 will be removed from the queue. 2 Service Points will be deducted from the user (update *User Summary* appropriately). An appropriate "Poor Service" message will popup indicating incident details (customer type, how many rounds they waited). An **incident record** is also stored to be added to the final performance report
- **At the end of each second round** (i.e. rounds 2, 4, 6, etc…), every diner who is remaining at their respective tables will become hungrier (deplete energy by 1).

The Ending

At the end of the last round, a performance summary will be printed to a **file**. This will be a .txt file and the **file name** must contain the user's name and date (e.g. "Fred_2021-11-08.txt"). The file will contain the following information:

- Date
- User's name
- Service Points
- Dishes Served
- Dishes Returned
- Number of diners fully fed.
- Number of diners suffering death from Poor Service
- Number of Ambroxians who met an untimely demise through 'death by curiosity'.
- Number of Customers lost from waiting too long.
- Poor Service Incident Records (listing of all Incident Records)

Marks

This assignment is worth **22%** of your final course mark. In addition to marks for code, overall marks will also be allocated for: Documentation/code formatting; User-friendly interface; Successful compilation; Correct execution; Exception classes.

Deductions will include:

- Violations of the Code Conventions. *Please ask if not sure.*
- Missing/inappropriate use of access specifiers.
- Inappropriate application of concepts.

Deliverables

1. **Assignment 2 is due for submission on Sunday 28th November 2021 by 11:55pm via the Moodle/eLearning submission tool.**
2. **Only ZIP files will be accepted.** No other compression types should be used. Failure to comply with this instruction will incur a **penalty**.
3. You must submit a **Plagiarism Declaration Form** with this assignment, if you do not accept the online one.

PLEASE NOTE: The specifications for this assignment are subject to change. You will be notified if any such changes were to occur.