

C16 - Lab:

Restaurant Menu Management System (File I/O and Exceptions)

Objective:

This lab focuses on creating a restaurant menu management system including file input/output operations and robust exception handling. The program will read restaurant and dish data from text files, enforce data validation rules, and use exceptions to manage error conditions.

Materials:

- Data sets available at <https://www.github.com/VMatosECC/Datasets/Restaurant>
- dishes.txt file (see format below)
- restaurants.txt file (see format below)

File Formats:

dishes.txt (Dish Name, Description, Calories, Price - Individual lines)

```
Spaghetti Aglio e Olio  
A simple yet flavorful dish...  
450  
12.99  
... (More dishes)
```

restaurants.txt: (Restaurant Name, Address, Phone, Location)

```
Delicious at Los Angeles  
555 Hollywood Blvd...  
(323) 123-4567  
Los Angeles  
... (More restaurants)
```

Instructions:

Part 1: Class Implementation (Restaurant, Dish, Menu)

1. Implement the Dish Class:

- Modify the Dish class constructor to throw a `std::runtime_error` if the calories are greater than 300.
- Keep the rest of the class implementation as before (name, description, calories, price, getters, display).

2. Implement the Menu Class:

- No changes are needed to the Menu class from the previous lab.

3. Implement the Restaurant Class:

- Modify the Restaurant class constructor to throw a `std::runtime_error` if the location (extracted from the address) is not one of the allowed cities ("Los Angeles", "Chicago", "Cleveland", "Miami").
- Keep the rest of the class implementation as before (name, address, phone, Menu, getters, addDish, removeDish, getMenu).

Part 2: File Input and Data Loading

1. Read Dishes from dishes.txt:

- Create a function `vector<Dish> loadDishesFromFile(string filename):`
 - Open the file for reading.
 - Read each line from the file.
 - Parse each line to extract the dish's name, description, calories, and price (use a delimiter like '|').
 - Create a Dish object.
 - Use a try-catch block to handle potential exceptions thrown by the Dish constructor (calories > 700).
 - If a Dish object is successfully created, add it to a `std::vector<Dish>`.
 - If an exception is caught, print an error message to `std::cout` indicating that the dish was not loaded (e.g., "Error: Dish 'Dish Name' not loaded: Too many calories.").
 - Close the file.
 - Return the `std::vector<Dish>`.
- Call this function in `main()` to load the dishes.

2. Read Restaurants from restaurants.txt:

- Create a function `vector<Restaurant> loadRestaurantsFromFile(string filename):`
 - Open the file for reading.
 - Read each line from the file.
 - Parse each line to extract the restaurant's name, address, phone, and location (use a delimiter like '|').
 - Create a Restaurant object.
 - Use a try-catch block to handle potential exceptions thrown by the Restaurant constructor (invalid location).
 - If a Restaurant object is successfully created, add it to a `std::vector<Restaurant>`.
 - If an exception is caught, print an error message to `std::cout` indicating that the restaurant was not loaded (e.g., "Error: Restaurant 'Restaurant Name' not loaded: Invalid location.").
 - Close the file.

- Return the `std::vector<Restaurant>`.
 - Call this function in `main()` to load the restaurants.
- 3. **Populate Menus (from file data):**
 - After loading the dishes and restaurants, iterate through the restaurants.
 - For each restaurant, add dishes to its menu. You can either:
 - Add all loaded dishes to every restaurant's menu (for simplicity).
 - Extend the file format to include which dishes belong to which restaurant, and implement the logic to add dishes accordingly.

Part 3: User Interaction and Queries

1. Implement the same user interaction and query features as in the previous lab:
 - Display a Restaurant's Information.
 - Display a Restaurant's Menu.
 - Search for a Dish.

Part 4: Error Handling and Exceptions

1. The core of the error handling is now in the file loading functions and constructors. Ensure that your `main()` function and user interaction parts are also prepared to handle potential exceptions (e.g., if a restaurant or dish is not found during a search). Use try-catch blocks where appropriate.

Example Code Snippets (Illustrative):

```
// Load Dishes from file
int main() {
    vector<Dish> dishes = loadDishesFromFile("c:/temp/restaurant/dishes.txt");
    vector<Restaurant> restaurants = loadRestaurantsFromFile("c:/temp/restaurant/restaurants.txt");

    // Assign all dishes to each restaurant (can be modified to be more selective)
    for (auto& restaurant : restaurants) {
        for (const auto& dish : dishes) {
            restaurant.addDish(dish);
        }
    }

    // Display restaurant information
    for (const auto& restaurant : restaurants) {
        restaurant.display();
    }

    cout << "\nAll done!" << endl;
}
```

Sample Output

Error: Dish 'Beef Bourguignon' not loaded: Too many calories.

Error: Restaurant 'Delicious at Springfield' not loaded: Invalid location.

Delicious at Los Angeles

555 Hollywood Blvd, Los Angeles, CA 90028

(323) 123-4567

Spaghetti Aglio e Olio - A simple yet flavorful dish with garlic, olive oil, and red pepper flakes.

(450 cal, \$12.99)

Classic Caesar Salad - Crisp romaine lettuce tossed with creamy Caesar dressing and croutons.

(320 cal, \$8.99)

Grilled Salmon with Asparagus - Flaky grilled salmon served with tender asparagus spears.

(550 cal, \$19.99)

Spanish Paella - A vibrant rice dish with seafood, chicken, and chorizo.

(680 cal, \$24.99)

Chicken Tikka Masala - Creamy and flavorful chicken curry with aromatic spices.

(610 cal, \$16.99)

Vegetarian Lasagna - Layers of pasta, ricotta cheese, and fresh vegetables in marinara sauce.

(580 cal, \$14.99)

Mushroom Risotto - Creamy Arborio rice cooked with savory mushrooms and Parmesan cheese.

(520 cal, \$15.99)

New York Style Cheesecake - Rich and creamy cheesecake with a graham cracker crust.

(480 cal, \$7.99)

Chocolate Lava Cake - Warm, decadent chocolate cake with a molten chocolate center.

(550 cal, \$8.99)

Delicious at Chicago

100 N Michigan Ave, Chicago, IL 60602

(312) 234-5678

Spaghetti Aglio e Olio - A simple yet flavorful dish with garlic, olive oil, and red pepper flakes.

(450 cal, \$12.99)

Classic Caesar Salad - Crisp romaine lettuce tossed with creamy Caesar dressing and croutons.

(320 cal, \$8.99)

Grilled Salmon with Asparagus - Flaky grilled salmon served with tender asparagus spears.

(550 cal, \$19.99)

Spanish Paella - A vibrant rice dish with seafood, chicken, and chorizo.

(680 cal, \$24.99)

Chicken Tikka Masala - Creamy and flavorful chicken curry with aromatic spices.

(610 cal, \$16.99)

Vegetarian Lasagna - Layers of pasta, ricotta cheese, and fresh vegetables in marinara sauce.

(580 cal, \$14.99)

Mushroom Risotto - Creamy Arborio rice cooked with savory mushrooms and Parmesan cheese.

(520 cal, \$15.99)

New York Style Cheesecake - Rich and creamy cheesecake with a graham cracker crust.

(480 cal, \$7.99)

Chocolate Lava Cake - Warm, decadent chocolate cake with a molten chocolate center.

(550 cal, \$8.99)

. . .

All done!