

## Review the key concepts covered during the activity.

### Key Points from the Activity

#### Stacks and Queues

- **Stack:** Adds and removes items from the top. Like a stack of books.
- **Queue:** Adds items at the back and removes from the front. Like a line at a store.

#### Making a Queue with Two Stacks

- **Enqueue:** Add customer to stackNewestOnTop.
- **Dequeue:** Move items to stackOldestOnTop when needed, then remove from the top of stackOldestOnTop.
- **Peek:** Look at the next customer without removing them.

#### Efficiency

Using two stacks makes sure adding and serving customers is fast overall.

#### Object-Oriented Programming (OOP)

- **Class Design:** TicketCounter class handles everything related to the queue.
- **Encapsulation:** Keeps the stacks hidden and uses methods to interact with them.

#### Error Handling

Methods check if the queue is empty and handle it nicely.

#### User Interaction

- **Menu Interface:** Simple text menu lets users choose what to do.
- **Input Handling:** Users can add, serve, peek, and see the queue status through menu options.

## Summary

I built a ticket system using two stacks to act like a queue. It covers basic concepts of stacks, queues, and OOP, and includes a simple menu for user interaction.

## Discuss the importance of queues in managing processes and scenarios like ticket counters.

### Importance of Queues

#### Managing Processes

- **Order:** Queues keep things in order. First come, first served. Just like waiting in line at a fast food place.
- **Fairness:** Everyone gets their turn based on when they arrived. No cutting in line!

#### Scenarios like Ticket Counters

- **Efficiency:** Helps manage crowds efficiently. Each person is served one by one without confusion.
- **Simplicity:** Easy to understand and follow. Customers know their spot in line.
- **Predictability:** You can estimate waiting time since it's clear who's next.

### Summary

Queues are essential for keeping processes smooth and fair, especially in places like ticket counters. They ensure everyone gets served in the order they arrive, making the system efficient and easy to manage.

<https://github.com/jims-sama/ticketingsystem>

