

## How can we implement a Queue using Stacks?

We specify the requirements as an unbounded Queue of integers and state that the students must implement the FIFO behaviour using only two unbounded integer Stacks (LIFO behaviour) to store the Queue contents.

The students are to compare the reliability of 2 different designs/solutions when the Stack components are unreliable.

You can implement your own designs, or use the 2 designs provided on next slides

## Design1:

The queue is specified as having two stack components which we will name as a *pushstack* and a *popstack*.

When a push request is made of the queue then this element is pushed directly onto the *pushstack*.

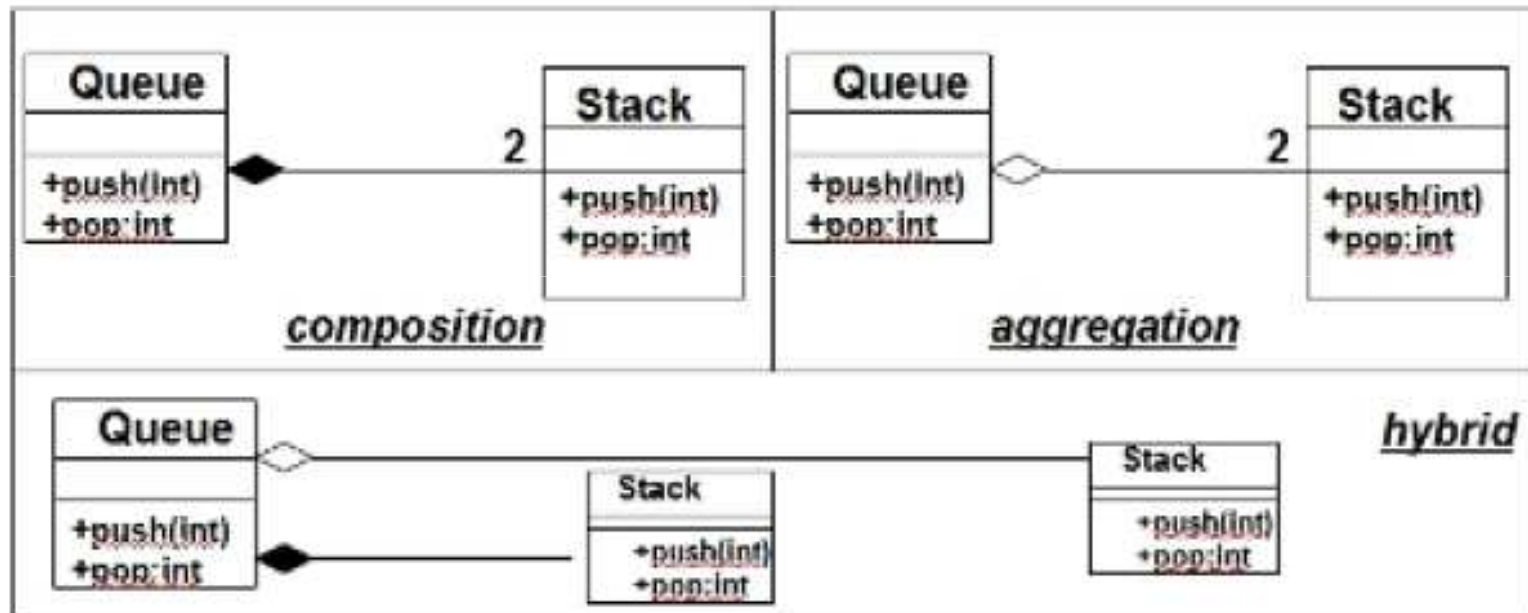
When a pop request is made of the queue then move all elements from the *pushstack* on to the *popstack* then pop off the last element of the *popstack* and then move all the elements back on to the *pushstack*.

## Design2:

The queue has two stack components which we will name as a *mainstack* and a *tempstack* and a boolean representing whether or not the *mainstack* is *ready to push*. (If it is not *ready to push* then we say that it is *ready to pop*).

When *ready to push*, if a push request is made of the queue then this element is pushed directly onto the *mainstack*, if a pop is requested then all the elements are moved from the *mainstack* to the *tempstack*, the *mainstack* and *tempstack* are swapped, the state is changed to *ready to pop* and the element popped off the *mainstack*.

## Design Questions: Aggregation or Composition



## Fault tolerance, robustness and reliability

If we know that the fundamental components can fail (following different failure patterns) then can the designs be analyzed in order to reason about the reliability of the system?

### Faulty Stacks (PBL)

Each Stack has a probability of being faulty for each of its public methods:

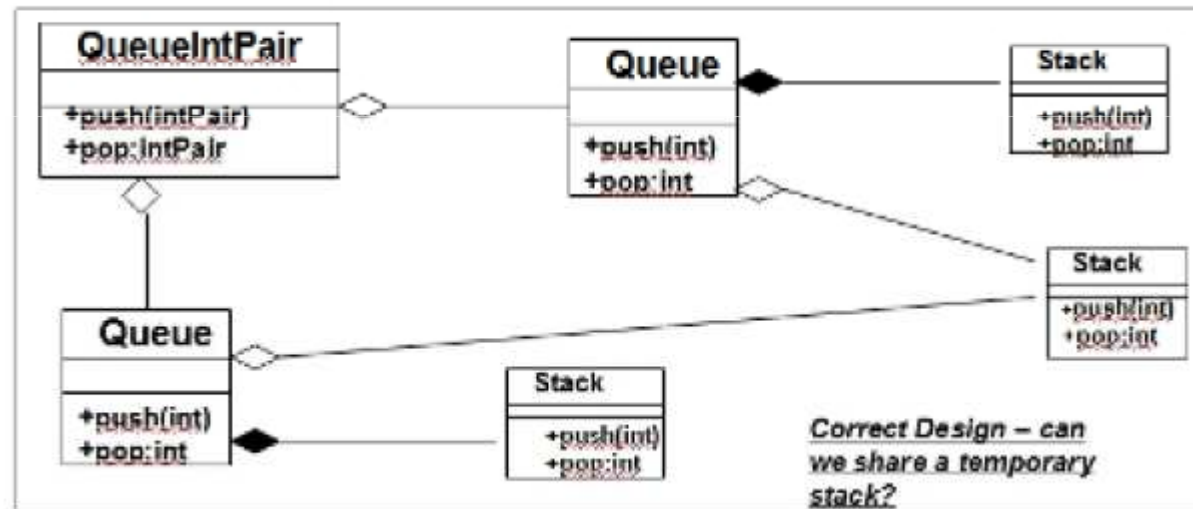
Push –  $1/p$       Pop –  $1/q$       Head –  $1/r$       IsEmpty – 0

When a fault occurs, an exception is thrown.

Analyze each of your designs in order to calculate the reliability of each of its methods

# Reliability Project

- 1) Re-analyze the 2 designs, if 1 and only 1 of the Stacks is unreliable.
- 2) You are to implement a Queue of integer pairs, using only unreliable Stacks. Follow the design, below. Is this design correct? If so, how reliable is it?



- 3) Propose a different design. Demonstrate its correctness and analyze its reliability