**Module 4: Stacks II & Queues**

Learning outcome:

* Familiarize with the Queue ADT using the List ADT using linked-list implementation
* To design and develop pointer-based Stack based on the given ADT
* Use the Queue in a typical application

Queues are a data structure that follows the rule of First In, First Out (FIFO). The linked list implementation of queues has the advantage of allowing to enqueue elements without a limited size.

Exercise 1: Queues

Suppose IIUM creates a Hajj programme for all staff members, which offers the opportunity for staff to fulfill their once in a lifetime Hajj requirement. However, there is an opening of only a limited number of staff in a season and staff will be offered on first-come, first served basis.

The LinkedQueue class which implements a pointer-based queue to allow enqueuing and dequeuing of pilgrims who joins the IIUM Hajj programme.

The LinkedQueue class should consists of the following attributes

* Staff number
* Staff name
* Faculty or division name

The class shall have the following member methods:

* Enqueuer
* Dequeuer
* Queue Displayer

Exercise 2: Linked-List Stacks

1. Use the previous definitions of theclass StaffList, to implements the stack data structure for objects of class Staff (as describe in the previous Tutorial Module). StaffList should include functions that perform Push and Pop operations.
2. **Write** StaffList class using the C++ programming language. Include error handling necessarily.
3. **Write and execute** a program to insert the same data in Exercise 1 in a StaffList object using Push operation.
4. **Display** the names and faculty/division of all elements in StaffList.