CONTENTS

Chapters and Topics **page**

CHAPTER 21 (DAY 4 Fore noon)

QUEUES

CHAPTER – 21

QUEUES

INTRODUCTION

Queues are a group of ordered elements like Stacks. Unlike stacks, the items are put one after the other so that always an item can be placed behind the previous item of a queue or an item can be removed from the beginning of the queue. There are two ends, one in the beginning and one at the end of the queue.

A ticketing window at a movie theatre or a sports event are the perfect example of a **queue**. New person enters the **queue** from the back and the next person to buy the ticket is called from the beginning of the **queue**. An assembly line is another example where parts are put in a sequence, new parts are sequenced behind the previous parts. Parts in the beginning of the **queue** are serviced next. If there are no parts in the **queue**, hence no servicing the parts. Similarly, when there are no persons in the **queue**, no tickets will be issued to anyone. Persons or parts are not allowed to be serviced from the middle or back of the **queue**, only the persons or parts in the beginning of a queue are serviced.

Similar example can be found at the recruitment center where application submitted first are processed first. New applications arrived will go to the bottom of the pile. Only the top item or item at the beginning will be processed next. Many such real life examples can be found, the elements of the queue may change but the basic operations remain the same.

Queues are also found at Bus Stops, Train stations, Banks, Cafeterias. In all the places, the next person joining the **queue** will enter at the end or rear of the line. Only the person in the beginning of the line is serviced, no person enters in the middle nor is any person from the middle of the line serviced.

QUEUE IMPLEMENTATION

A **queue** can be implemented using an array of elements in which case we need to know how many items we can put in a **queue** at a time. That is the queue can have at the most the defined maximum number of elements which is implementation dependent. Once we declare the maximum elements, it stays for the entire duration until we compile the code with the new maximum set value. The items that can be **queued** are integers, characters, floats, or any user defined data types like structures. When no items are placed in the **queue**, the **queue** is in a **empty** state. While the **queue** is empty no items can be removed from **the queue**, but items can be added to an **empty queue**.