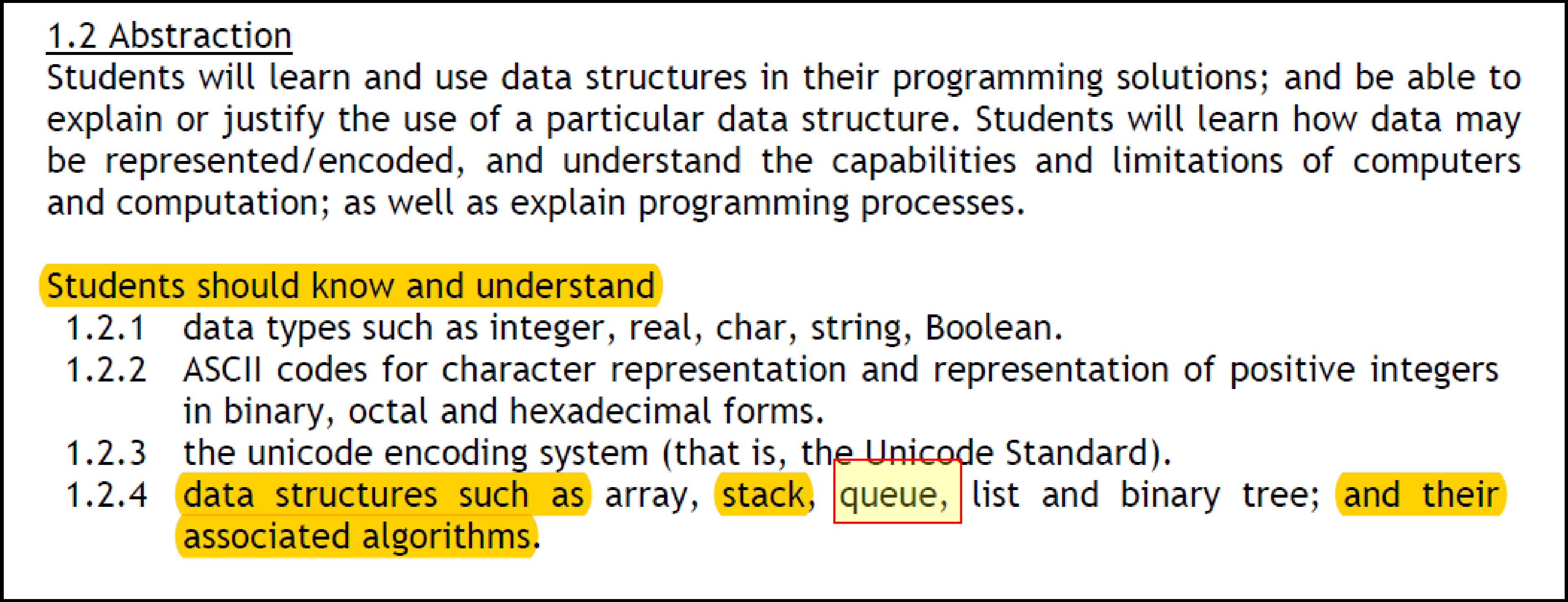
LT9b QUEUE

Learning Outcomes



1.2 Abstraction

Students will learn and use data structures in their programming solutions; and be able to explain or justify the use of a particular data structure. Students will learn how data may be represented/encoded, and understand the capabilities and limitations of computers and computation; as well as explain programming processes.

Students should know and understand

1.2.1 data types such as integer, real, char, string, Boolean.

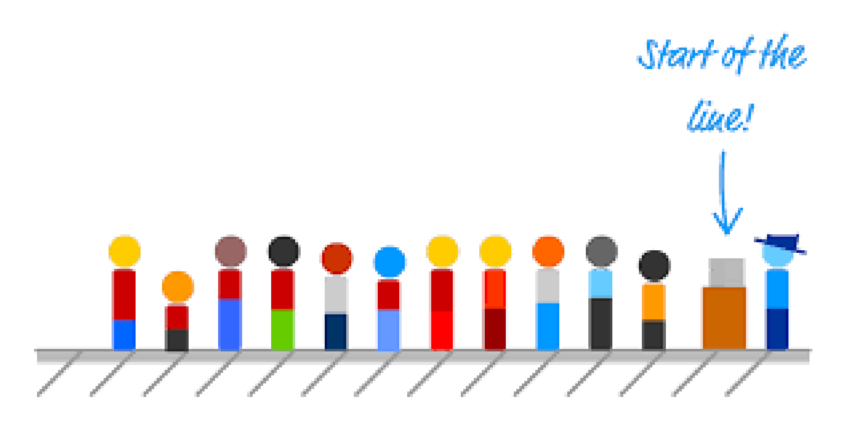
1.2.2 ASCII codes for character representation and representation of positive integers in binary, octal and hexadecimal forms.

1.2.3 the unicode encoding system (that is, the Unicode Standard).

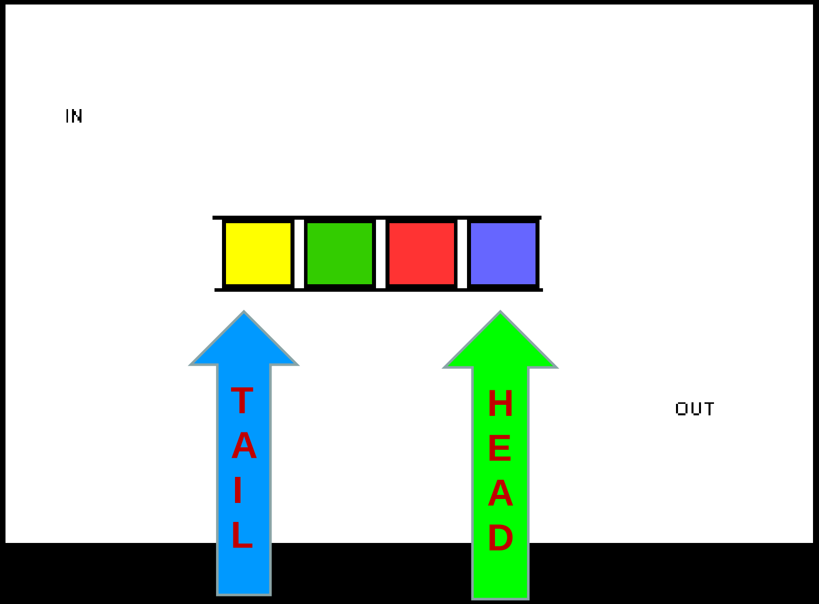
1.2.4 data structures such as array, stack, queue, list and binary tree; and their associated algorithms.

What is a queue?

- A queue is an ordered collection of items.
- It is a **first in first out (FIFO)** data structure.
- Elements are usually removed in the same order as which they are added.



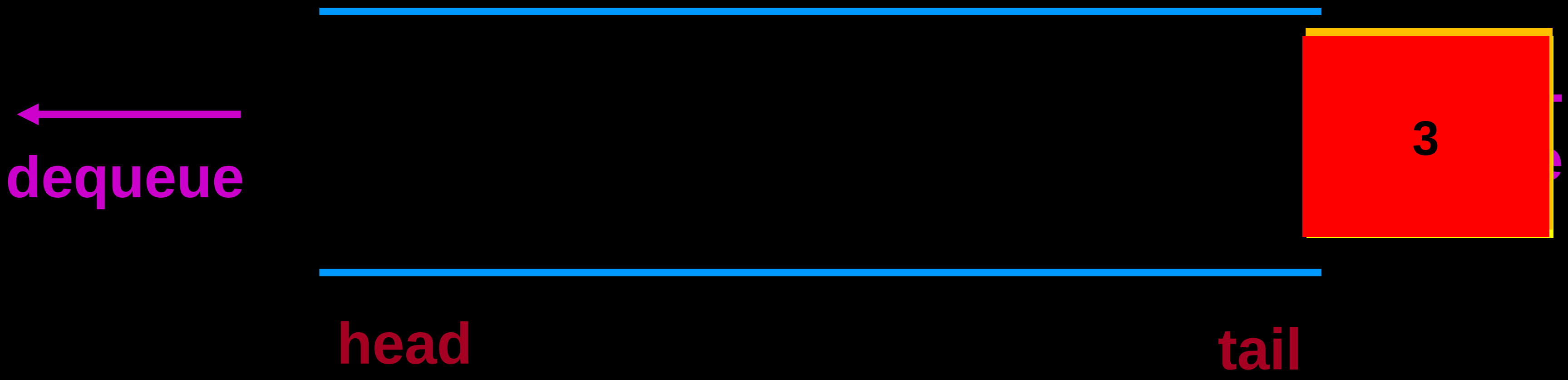
What is a queue?



An element is **added** to the **tail** position of the queue.

An element is **removed** from the **head** of the queue.

Implementation of a FIFO queue



dequeue

3

head

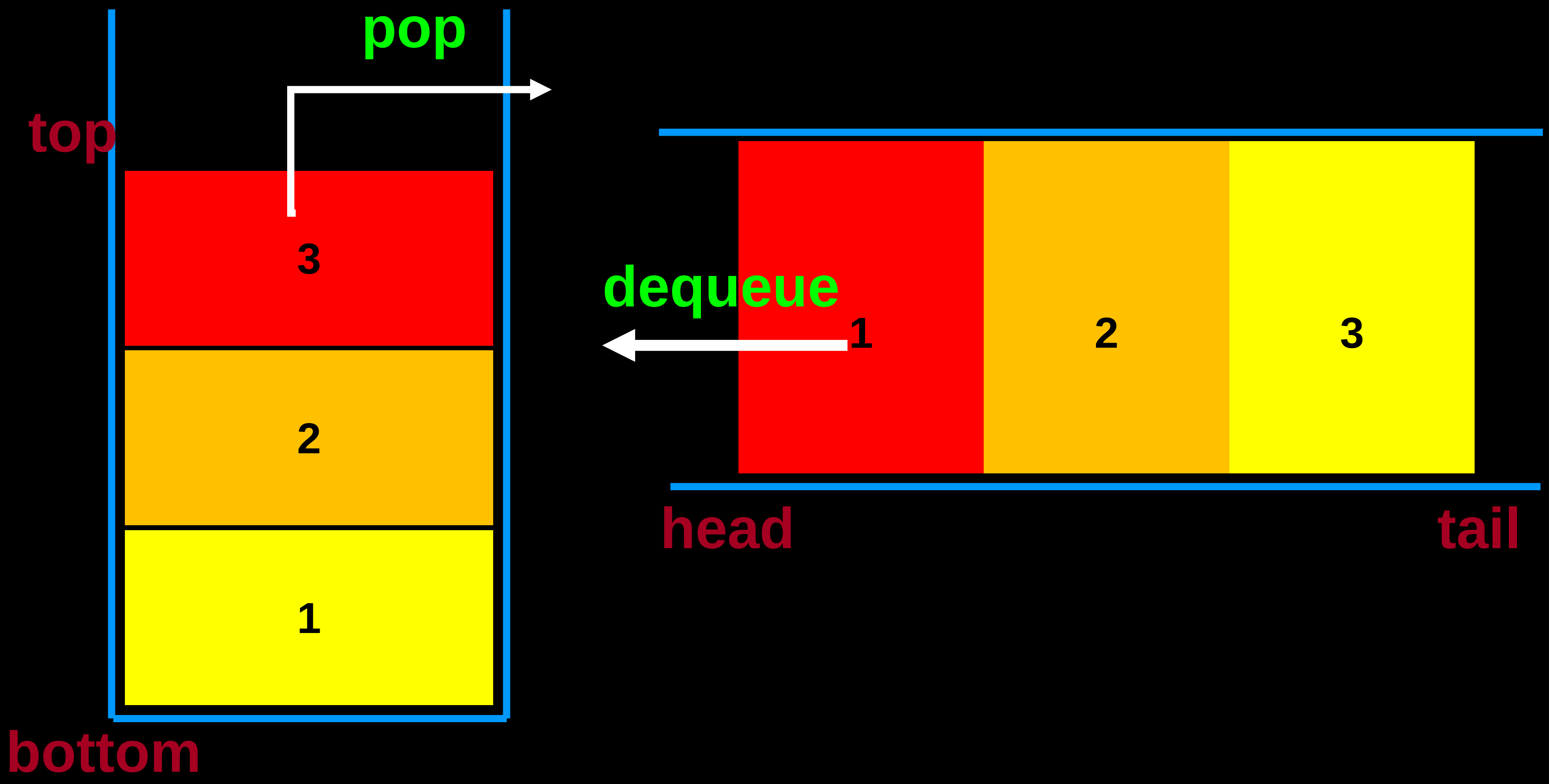
tail

Main Functions

enqueue / push

dequeue / pop

What’s the difference?



Stack vs Queue

You will be implementing the queue ADT in lecture training LT9b:

Queue ADT	Description
`make_queue()`	returns a new, empty queue
`enqueue(queue,item)`	adds item to queue
`dequeue(queue)`	removes the first item from queue and returns it
`front(queue)`	returns the first object of the queue but does not remove it from the queue
`is_empty(queue)`	returns True if queue is empty, False if otherwise

List Operations for Queue

Queue Operation	**List Operation**	**Description**
enqueue	lst.append(item)	Modifies list by adding an item to the end of the list
dequeue	lst.pop(i)	Removes the item at index i and returns it

Implement a FIFO queue using a list

```
q = make_queue()

dequeue(q)          →  None  : empty queue,
enqueue(q, 5)       nothing to dequeue
enqueue(q, 3)
dequeue(q)          →  5
dequeue(q)          →  3
is_empty(q)         →  True
```

Applications of Queue (Tutorial)

- Scheduling of processes
- Network printer queue
- Keyboard buffer

Summary

- A queue is an ordered collection of items.
- Used when the order needs to be preserved
- A queue is FIFO data structure
- Typical operations include enqueue (add items) and dequeue (remove items)

Stack vs Queue

STACK	QUEUE
Follows the LIFO order	Follows the FIFO order
Operations are push and pop	Operations are enqueue and dequeue
pop removes an item from the top of the stack	dequeue removes an item from the head of the queue
peek retrieves an item from the top of the stack	front retrieves an item from the head of the queue
A stack is a vertical collection of items	A queue is a horizontal collection of items