CSE 3010 – Data Structures & Algorithms

Lecture #22

What will be covered today

- Mid-semester examination pattern
- Implementation of queue data structure
  - Using singly linked list

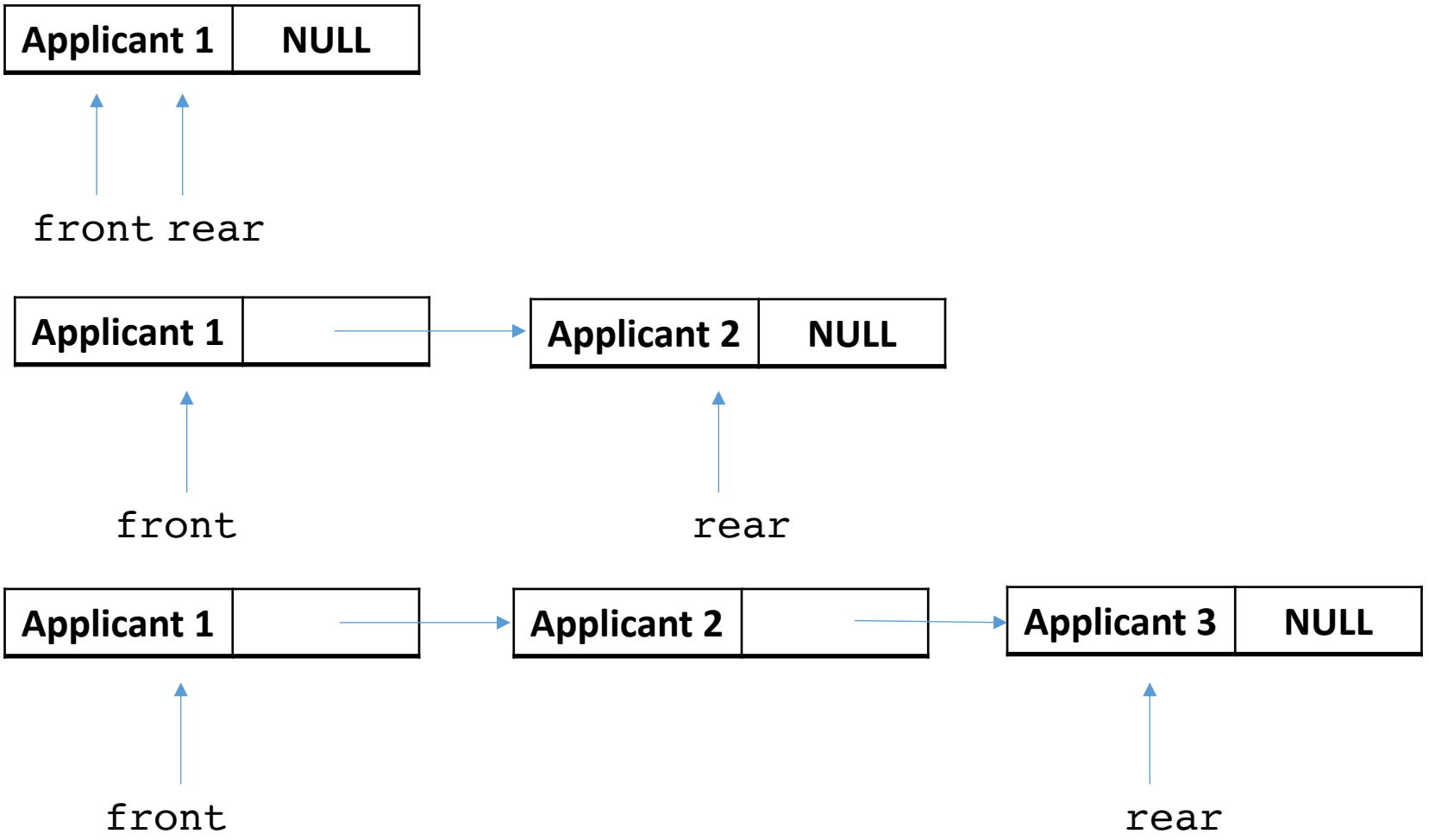
Mid-semester examination pattern

Date of Examination	27th January 2020
Time	9:00 am
Duration	2 hours
Format	Open book
Type	Problem solving
Through	LMS
Syllabus	Until **Queue data structure**

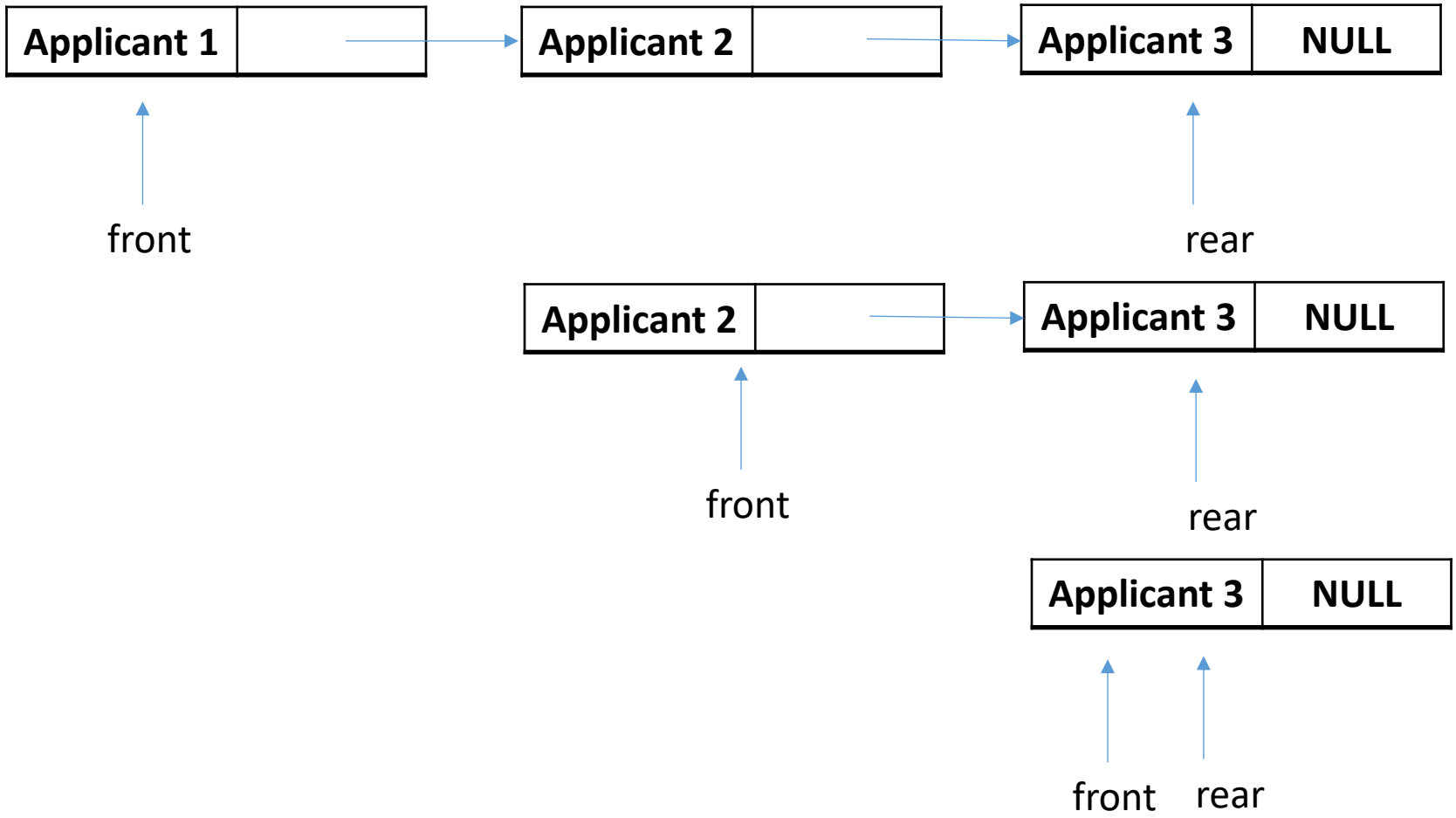
Implementation of Queue using singly linked list – When an item is added to the queue

`front` and `rear` pointers of a queue are null when a queue is created

front = rear = NULL



Implementation of Queue using singly linked list – When an item is deleted from the queue



`front` and `rear` are NULL when a queue is empty

Operations on a singly linked list for Queue

```
QNODE* createNode(ITEM item) {

    QNODE *temp = (QNODE*) malloc(sizeof(QNODE));
    // Check for temp = NULL for full condition
    temp->item = item;
    temp->next = NULL;
    return temp;
}

// To create an empty queue
QUEUE* createQueue() {
    QUEUE *queue = (QUEUE*) malloc(sizeof(QUEUE));
    queue->front_queue = NULL;
    queue->rear_queue = NULL;
    return queue;
}
```

Operations on a singly linked list for Queue

```
int add(QUEUE *queue, ITEM item) {

    QNODE *temp = createNode(item);
    // Check for temp = NULL for full condition
    if (queue->rear_queue == NULL) {
        queue->front_queue = temp;
        queue->rear_queue = temp;
        return 1;
    }
    else {
        queue->rear_queue->next = temp;
        queue->rear_queue = temp;
        return 1;
    }
}
```

Operations on a singly linked list for Queue

```
ITEM delete(QUEUE *queue) {
    ITEM tempItem;
    if (isEmpty(queue))
        tempItem.appln_name[0] = '\0';
    else {
        QNODE *tempNode;
        tempNode = queue->front_queue;
        tempItem = queue->front_queue->item;
        queue->front_queue = queue->front_queue->next;
        free(tempNode);
     if (queue->front_queue == NULL)
            queue->rear_queue = NULL;
    }
    return tempItem;
}
```