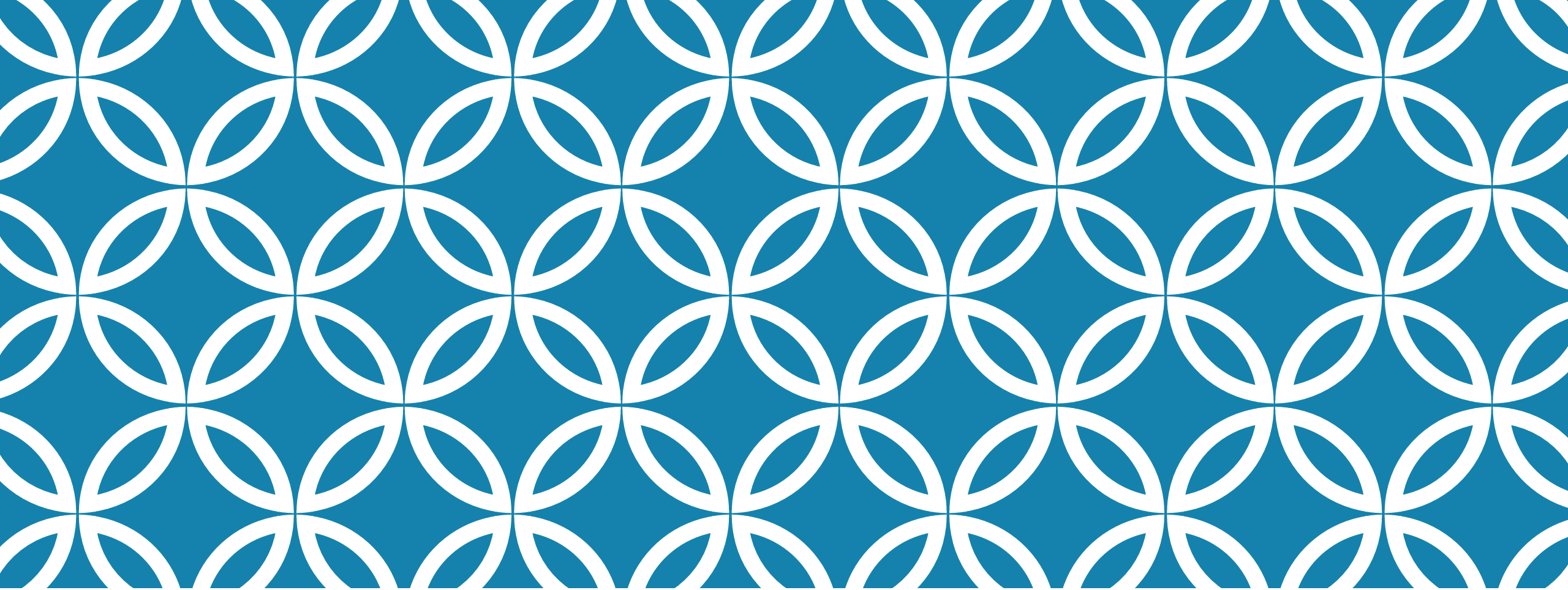


COMP 2611, Data Structures

LECTURE 1: OVERVIEW OF COURSE & REVIEW OF LINKED LISTS

A DATA STRUCTURE

- What are the basic operations performed on data?

Read Write

- A data structure is a way of storing data in a computer so that it can be efficiently retrieved, modified, etc.

- A well-designed data structure enables operations such as reading and writing data to be performed, minimizing execution time and memory space.

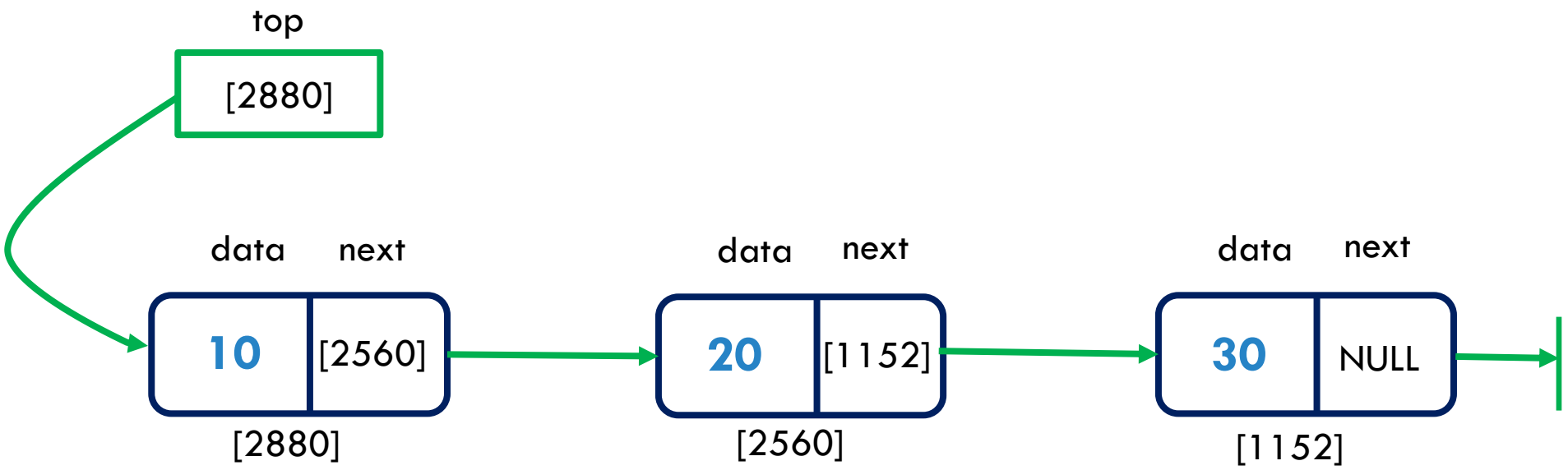
DATA STRUCTURES TO BE COVERED

- Linked lists, stacks, queues (review)
- Binary trees and binary search trees
- Heaps
- Priority queues
- Graphs
- Hashtables
- Matrices with special properties

ALGORITHMS

- Heapsort
- Mergesort
- Quicksort

REVIEW OF LINKED LIST DATA STRUCTURE



- *data* could be a simple type such as int, double or char
- *data* could be a struct

NODE IN A LINKED LIST

```
struct Node {
    int data;
    Node * next;
};
```

In *createNode* function:

```
Node * newNode;
newNode = new Node;
```

CREATENODE FUNCTION

```
Node * createNode (int n) {
    Node * newNode;
    newNode = new Node;
    newNode->data = n;
    newNode->next = NULL;

    return newNode;
}
```

INSERTING NODES IN A LINKED LIST

A node can be inserted:

- At the top of the linked list
- At the end of the linked list
- Somewhere between the top and the end of the linked list

We will now look at a few examples where nodes are inserted at the top of a linked list.