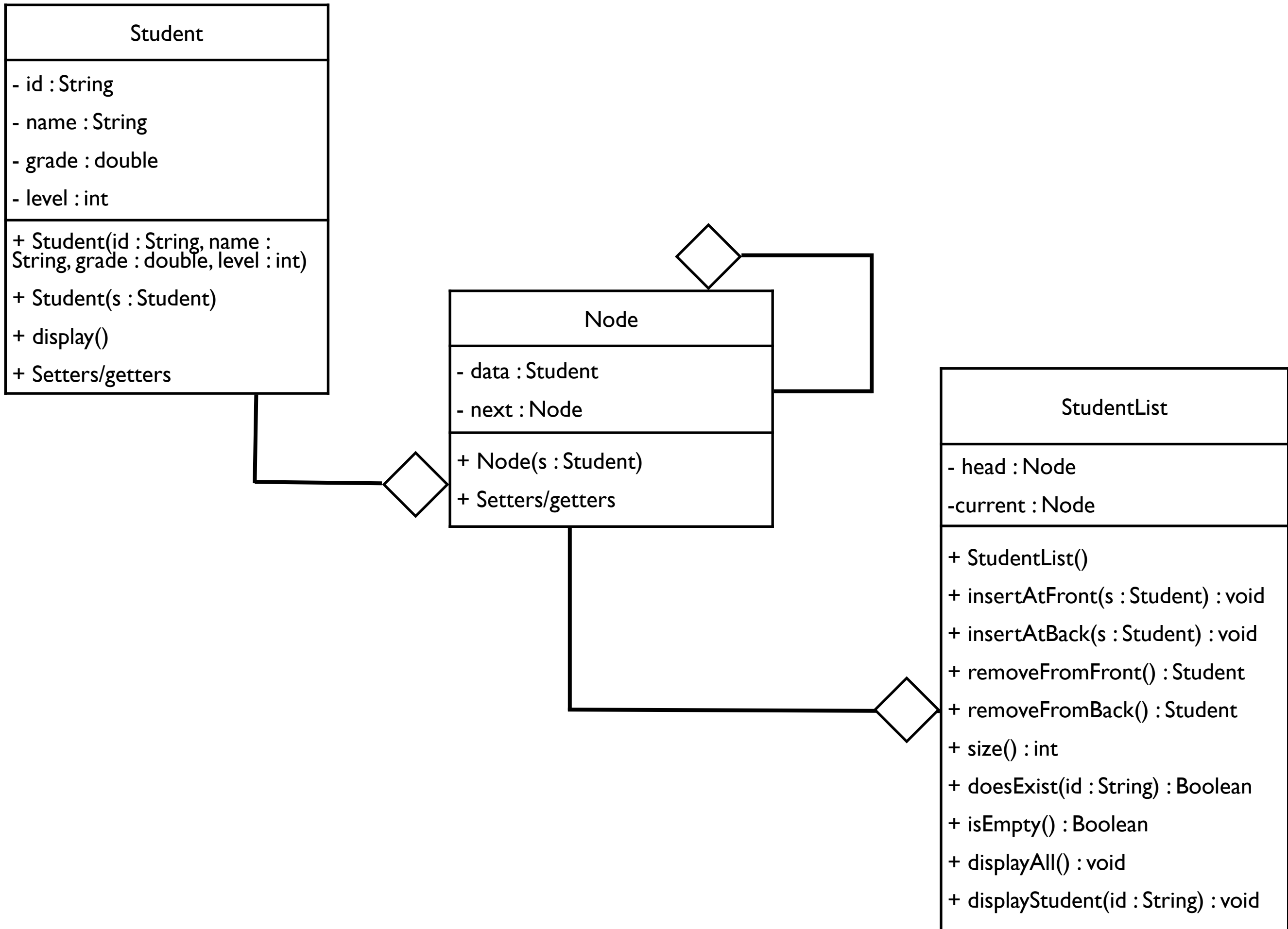


CSC 113

Tutorial 11

Linked List



Class Student

```
public class Student {
```

```
    private String id;  
    private String name;  
    private double gpa;  
    private int level;
```

```
    public Student()  
    {}
```

```
    public Student(String name, String id, double gpa, int level)  
    {  
        this.id = id; this.name = name;  
        this.gpa = gpa; this.level = level;  
    }
```

```
    public Student(Student s)  
    {  
        id = s.id; name = s.name;  
        gpa = s.gpa; level = s.level;  
    }
```

```
    public void display()  
    {  
        System.out.println("ID: "+id);  
        System.out.println("Name: "+name);  
        System.out.println("level: "+level);  
        System.out.println("GPA: "+gpa);  
    }
```

Student
- id : String - name : String - grade : double - level : int
+ Student(id : String, name : String, grade : double, level : int) + Student(s : Student) + display() + Setters/getters

Class Node

```
public class Node {  
    private Student data;  
    private Node next;  
  
    public Node(Student s)  
    {  
        data = s;  
        next = null;  
    }  
}
```

Node
- data : Student - next : Node
+ Node(s : Student) + Setters/getters

Class StudentList

```
public class StudentList {
    private Node head, current;

    public StudentList() { head = current = null; }

    public void insertAtFront(Student s)
    {
        Node n = new Node(s);
        n.setNext(head);
        head = n;
    }

    public void insertAtBack(Student s)
    {
        Node n = new Node(s);

        if(isEmpty()) { insertAtFront(s); return; }

        current = head;
        while(current.getNext() != null)
        { current = current.getNext(); }

        current.setNext(n);
    }
}
```

Insert

StudentList
- head : Node -current : Node
+ StudentList() + insertAtFront(s : Student) : void + insertAtBack(s : Student) : void + removeFromFront() : Student + removeFromBack() : Student + size() : int + doesExist(id : String) : Boolean + isEmpty() : Boolean + displayAll() : void + displayStudent(id : String) : void

Class StudentList

```
public Student removeFromFront()
{
    Node n = head;
    head = head.getNext();
    return n.getData();
}

public Student removeFromBack()
{
    Student result;
    Node n = head;
    current = head;

    if(isEmpty()) return null;

    while(current.getNext() != null)
    {
        n = current;
        current = current.getNext();
    }
    result = current.getData();

    if(size() == 1) head = null;
    else n.setNext(null);

    return result;
}
```

Remove

StudentList
- head : Node -current : Node
+ StudentList() + insertAtFront(s : Student) : void + insertAtBack(s : Student) : void + removeFromFront() : Student + removeFromBack() : Student + size() : int + doesExist(id : String) : Boolean + isEmpty() : Boolean + displayAll() : void + displayStudent(id : String) : void

Class StudentList

size, doesExist

```
public boolean doesExist(String id)
{
    current = head;

    while(current != null)
    {
        if(current.getData().getId().equals(id))
            return true;
        current = current.getNext();
    }

    return false;
}

public int size()
{
    int count = 0;
    current = head;

    while(current != null)
    {
        count++;
        current = current.getNext();
    }

    return count;
}
```

StudentList

- head : Node

-current : Node

+ StudentList()

+ insertAtFront(s : Student) : void

+ insertAtBack(s : Student) : void

+ removeFromFront() : Student

+ removeFromBack() : Student

+ size() : int

+ doesExist(id : String) : Boolean

+ isEmpty() : Boolean

+ displayAll() : void

+ displayStudent(id : String) : void

Class StudentList isEmpty, displayAll

```
public boolean isEmpty()  
    { return head == null ; }
```

```
public void displayAll()  
    {  
        current = head;  
        System.out.println("-----");  
        while(current != null)  
        {  
            current.getData().display();  
            current = current.getNext();  
        }  
    }
```

StudentList
- head : Node -current : Node
+ StudentList() + insertAtFront(s : Student) : void + insertAtBack(s : Student) : void + removeFromFront() : Student + removeFromBack() : Student + size() : int + doesExist(id : String) : Boolean + isEmpty() : Boolean + displayAll() : void + displayStudent(id : String) : void

Class StudentList

displayStudent

```
public void displayStudent(String id)
{
    current = head;

    while(current != null)
    {
        if(current.getData().getId().equals(id))
        {
            current.getData().display();
            return;
        }
        current = current.getNext();
    }

    System.out.println("Student with ID "+id+" does not exist.");
}
```

StudentList
- head : Node -current : Node
+ StudentList() + insertAtFront(s : Student) : void + insertAtBack(s : Student) : void + removeFromFront() : Student + removeFromBack() : Student + size() : int + doesExist(id : String) : Boolean + isEmpty() : Boolean + displayAll() : void + displayStudent(id : String) : void

Main

```
StudentList sl = new StudentList();
```

```
System.out.println("\nsize = "+sl.size());  
sl.insertAtFront(new Student("ali", "123", 3, 3));  
sl.displayAll();  
sl.insertAtFront(new Student("majed", "456", 4, 4));  
sl.displayAll();  
sl.insertAtBack(new Student("bandr", "789", 5, 4));  
sl.displayAll();
```

```
System.out.println("\nsize = "+sl.size());  
System.out.println("Does 123 exist? "+sl.doesExist("123"));  
System.out.println("Does 321 exist? "+sl.doesExist("321"));
```

```
System.out.println("\nPrint 456: "); sl.displayStudent("456");  
System.out.println("\nPrint 654: "); sl.displayStudent("654");
```

```
sl.removeFromFront();  
sl.displayAll();  
sl.removeFromBack();  
sl.displayAll();
```

Main output

```
size = 0
-----
ID: 123
Name: ali
level: 3
GPA: 3.0
-----
ID: 456
Name: majed
level: 4
GPA: 4.0
ID: 123
Name: ali
level: 3
GPA: 3.0
-----
ID: 456
Name: majed
level: 4
GPA: 4.0
ID: 123
Name: ali
level: 3
GPA: 3.0
```

```
ID: 789
Name: bandr
level: 4
GPA: 5.0
```

```
size = 3
Does 123 exist? true
Does 321 exist? false
```

```
Print 456:
ID: 456
Name: majed
level: 4
GPA: 4.0
```

```
Print 654:
Student with ID 654 does not
exist.
```

```
-----
ID: 123
Name: ali
level: 3
GPA: 3.0
ID: 789
```