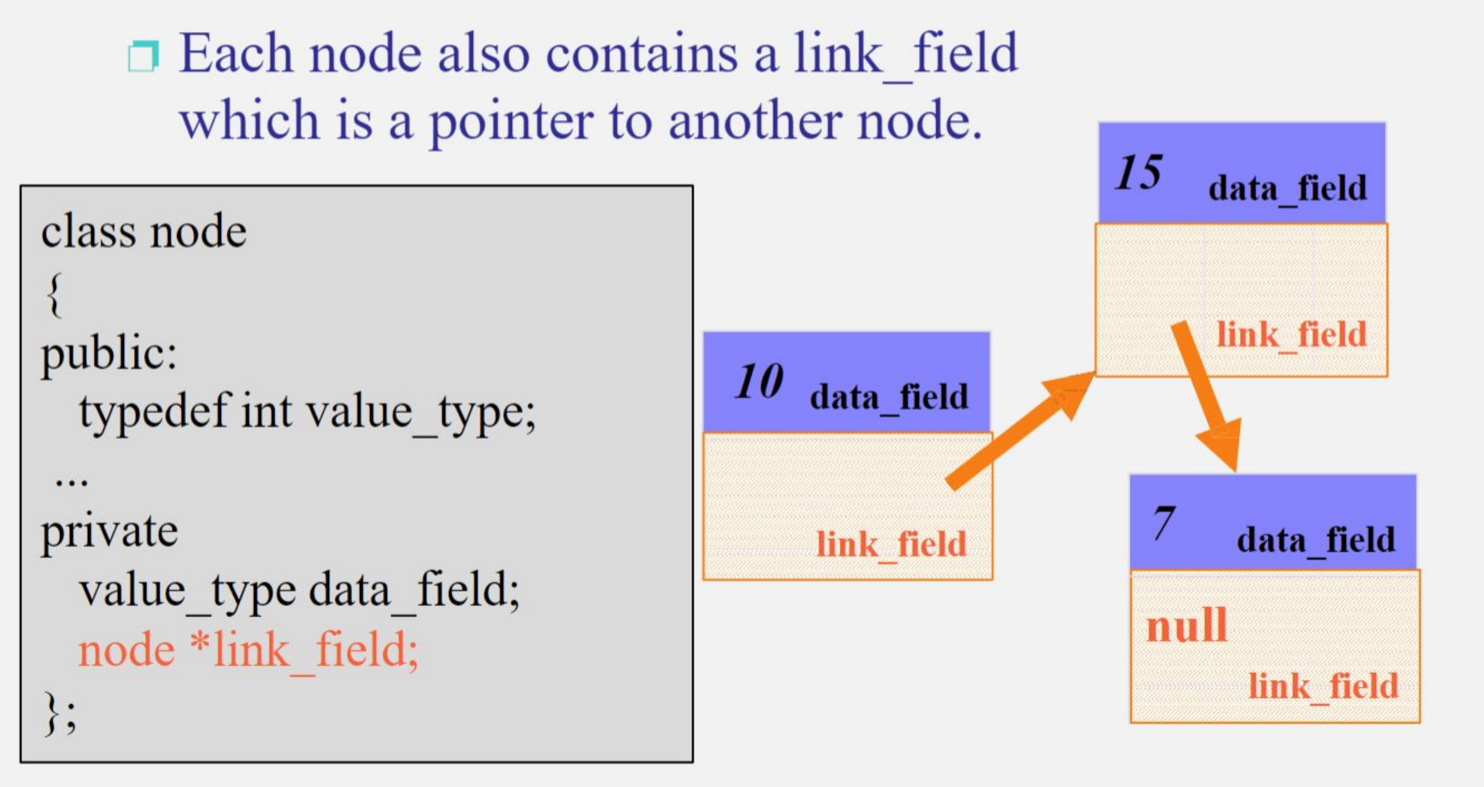
Lab 6

Table of Contents

- Review Linked List
- Lab 6 Description

Linked List in C++

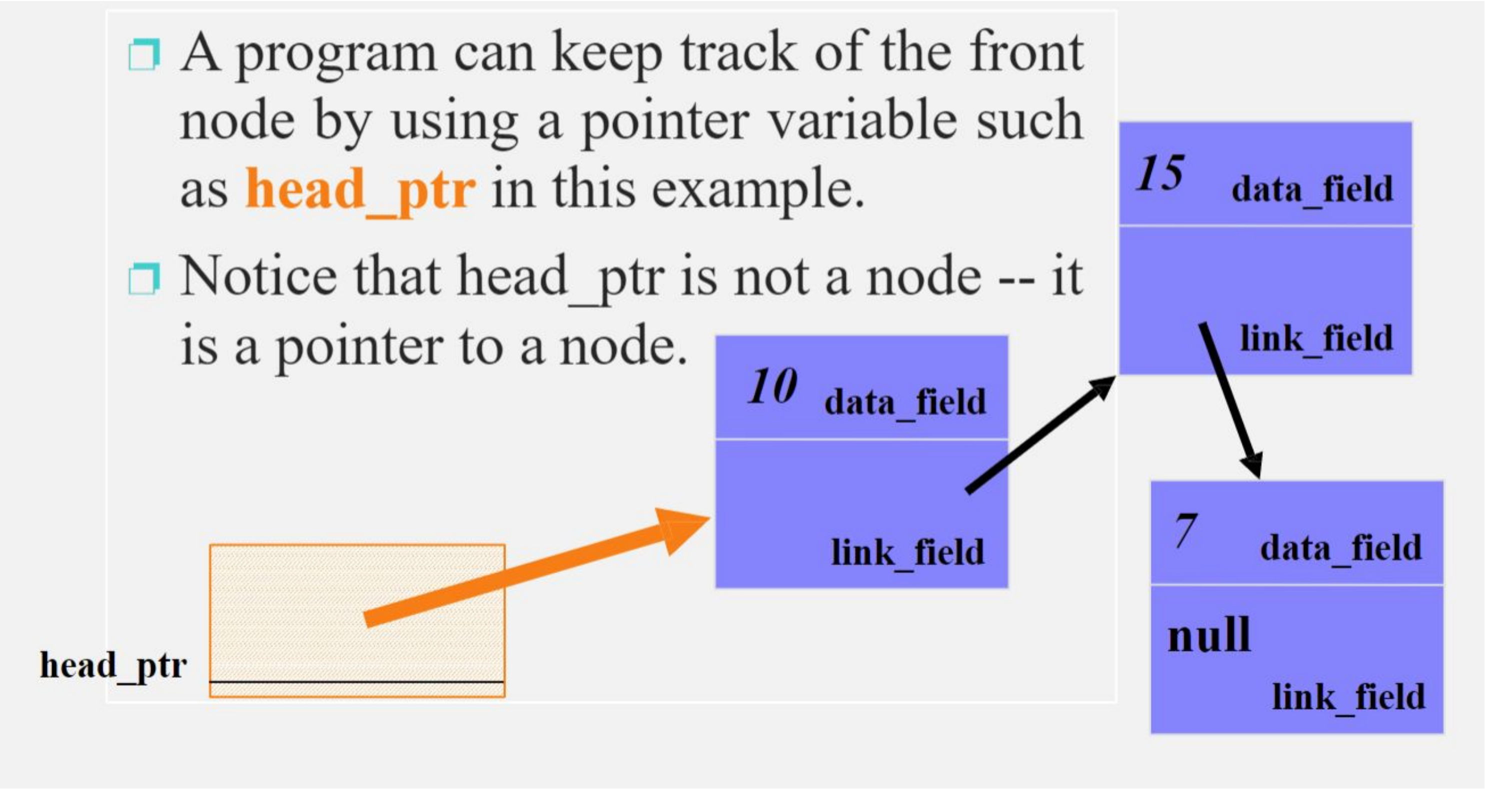
From Lecture Slide



- Each node also contains a link_field which is a pointer to another node.

```
class node
{
public:
   typedef int value_type;
 ...
private
   value_type data_field;
   node *link_field;
};
```

From Lecture Slide



- A program can keep track of the front node by using a pointer variable such as **head_ptr** in this example.
- Notice that head_ptr is not a node -- it is a pointer to a node.

Lab 6 Description

Project Structure

- lab6sampleinput.txt
- Roster
  - roster.h
  - roster.cxx
  - rostermain.cxx
  - node.h (optional)
- Makefile

Project Structure

- Build upon Lab3
- Functions to be added
  - Node Class / Structure
  - Hint: think about our scenario, how can we make insertion O(1)?
- Functions to be modified (or not)
  - Constructor
  - Destructor
  - insert
  - erase
  - (Add more functions if you would like to)
- Test Plan needs to be updated
  - What should change

roster.h (Class Containers)

- Two Classes
  - Student (regular class)
  - Roster (class container)
  - Node (structure or class)
    - think about where this should be: in another class, or used globally?
    - can be in a separate file

rostermain.cxx (rostertest)

- Main Function For Test
  - takes user input from input file
    - `rosterTest < input_file > outout_file` takes care of it
    - Or you can use ifstream
  - reads line by line
    - already done in sample main
  - format for line input
    - <command key> <params (depend on key)>
    - <command key> <params> format
      - <A> <ID> <first> <last> – insertion of a student with the input ID, first, last
        - ex: A 1234567 John Doe
      - <X> <ID> – deletion of student with ID
        - ex: X 7654321
      - “L” – list roster of students
        - ex: L

rostermain.cxx testing tips (rostertest)

- Edge Cases
  - insertion
    - under what cases we can not insert a student into the roster?
  - deletion
    - under what cases we can not delete a student into the roster?
  - list
    - under what cases we can not list the roster?

Test Plan

- Talk about the expected difference between lab 4 and lab 6
- Which one is better?
    - in term of efficiency, complexity, and flexibility
    - Hint: try to find a way to record the time it takes for the program to run (sounds familar) and compared to lab 3 and lab 4

Deliverables

- All .cpp files
- All .h files
- Makefile
- output file
- test plan

Demo

- Test Plan
- Code compilation/run

Other Tips

- Test code frequently
- Test your code comprehensively
  - Think about what needs to be tested
  - Points will be deducted if you missed critical test cases

Don’t Forget

- Submit the code before next week’s deadline
- File with guide to implement and hints are in Camino
    - Make sure your code can run on school Linux server