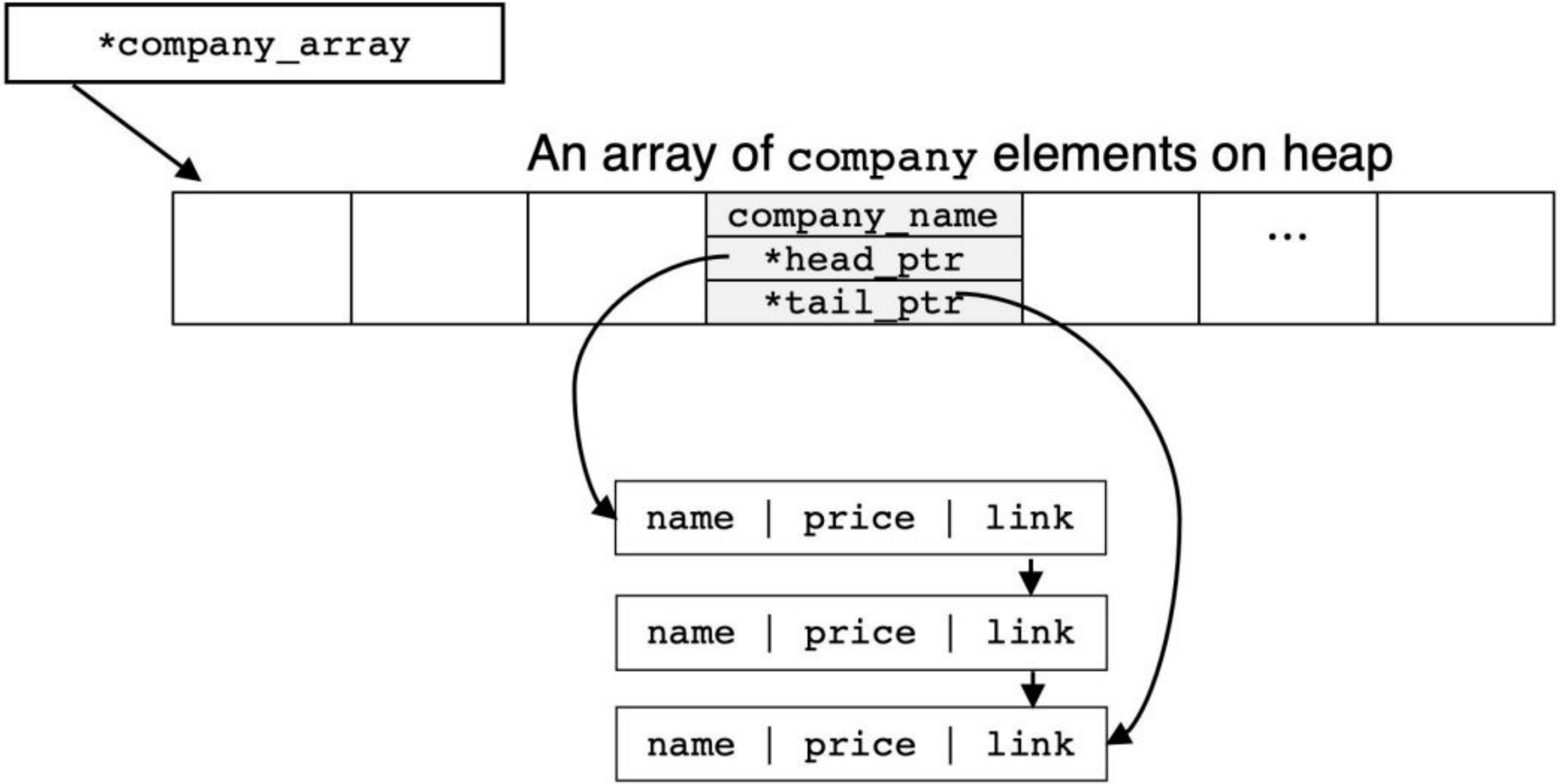
Lab 7


*company_array
An array of company elements on heap
company_name
*head_ptr
*tail_ptr
...
name | price | link
name | price | link
name | price | link

Database

Node

- One node of a linked list
  - One product and link to next node
- Once again, there are toolkit functions built into the implementation of this class that can make your **company** easier
- Private Variables
  - std::string name; //name of product
  - float price; //price of product
  - node* link // link to next product
- Functionality similar to **node** class from last week

Company

- Maintains a linked list of products (**node** class)
  - NOT allowed to have duplicate products in a company
- Forward linked list
  - Order does NOT matter
- Private Variables
  - std::string company_name;   //name of company
  - node *head_ptr;   //head of linked list of products
  - node* tail_ptr   // tail of linked list of products

Database

- Stores the name of companies as well as their products
  - NOT allowed to have duplicate companies in a database
- Builds off of the **company** and **node** classes
- Private Variables
  - company *company_array;
  - size_t aloc_slots;
  - size_t used_slots;
- Functionality similar to **myString** class from lab 5

Provided Files

- Node
  - node.h
  - node.cpp
- Company
  - company.h
  - company.cpp
- Database
  - database.h
  - database.cpp
- Example_expected_output.rtf

Don’t Forget

- Demo code to me
  - Today or next week
  - Must compile and run on linux servers
- Submit your code before week’s deadline
- Comment code
- File with description of lab is on Camino
  - Submission guidelines