Linked Lists

- Reading: Savitch, Chapter 10

Objectives

- To learn
  - what is a linked list
  - how to declare a linked list in JAVA
  - how to create and traverse a linked list

Motivation

- Write a program to store information about a class of students (eg., names, results, etc)
- Problems associated with use of an array
  - how big should the array be?
  - too small => ?
  - too big => ?
- Other examples: address book, dictionary

Dynamic structures

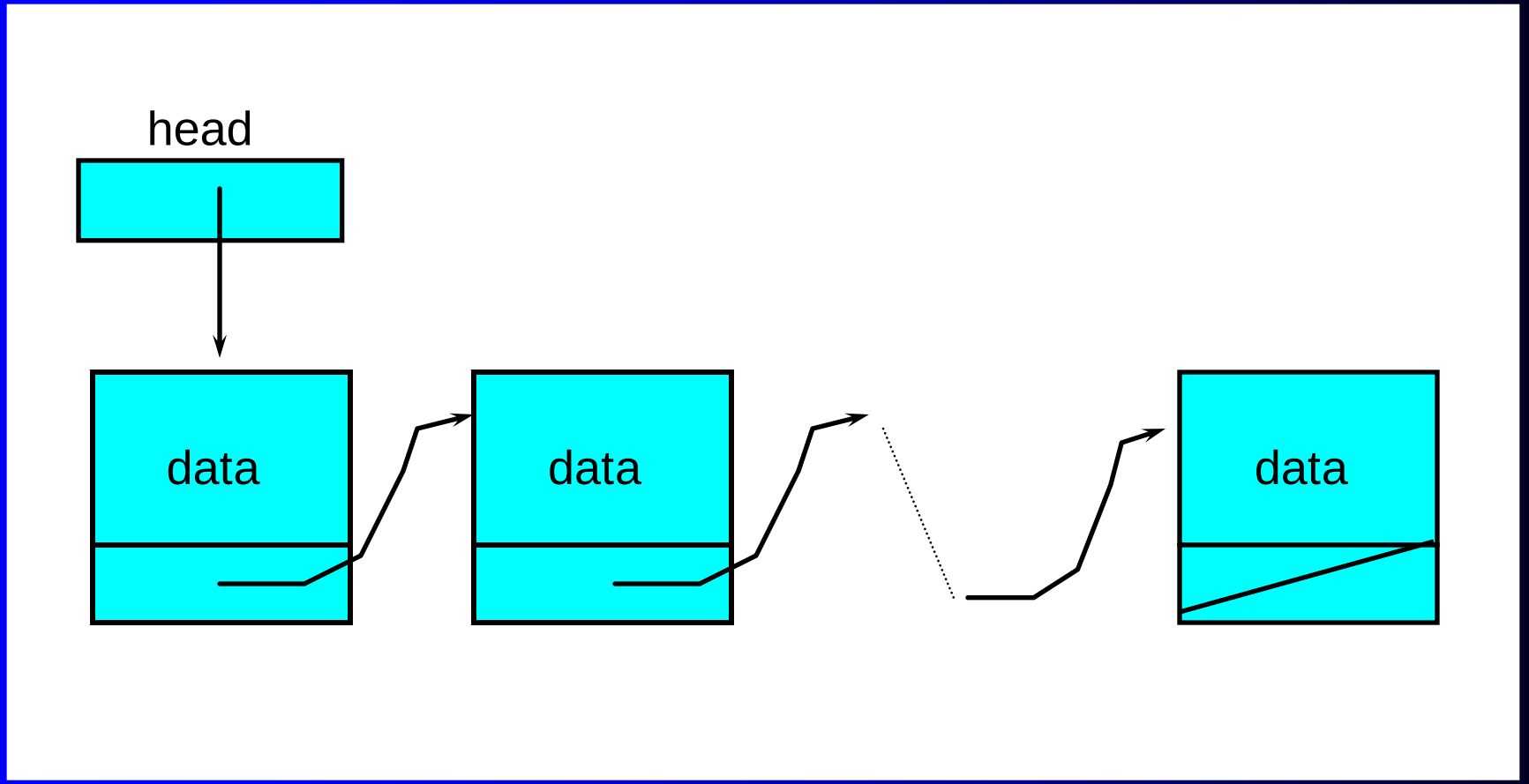
- Grow and diminish dynamically based on the program requirement.

Linked List

- What is a linked list?

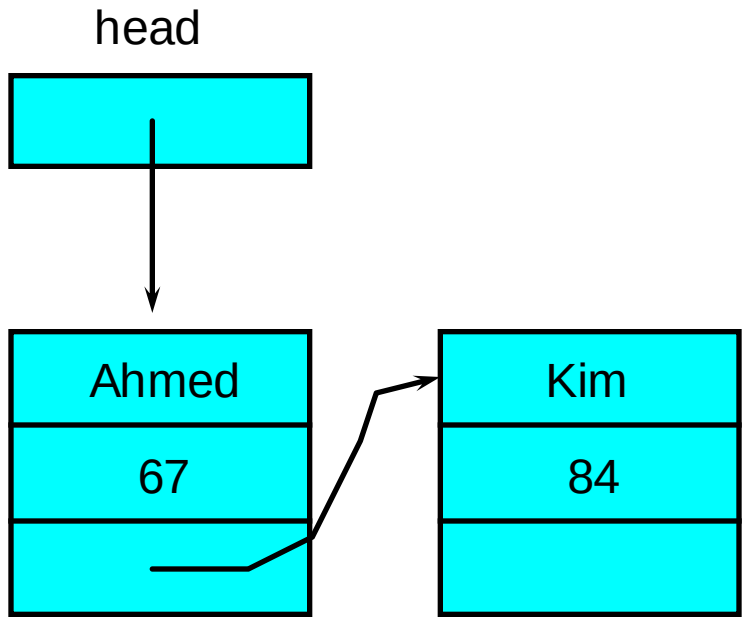
  a variable-length collection of objects (of the same class). Each object is called a node of the linked list. Each node contains a reference to the next node.

A linked list



Linked list declaration

```
class StudentNode {
   private String name;
   private int mark;
   private StudentNode next;
   public StudentNode(String _n, int _m)
   {  name = _n;  mark = _m;
      next = null;
    }
   ... ...
}

class LinkedList {
   private StudentNode head = null;
   ... ...
}
```



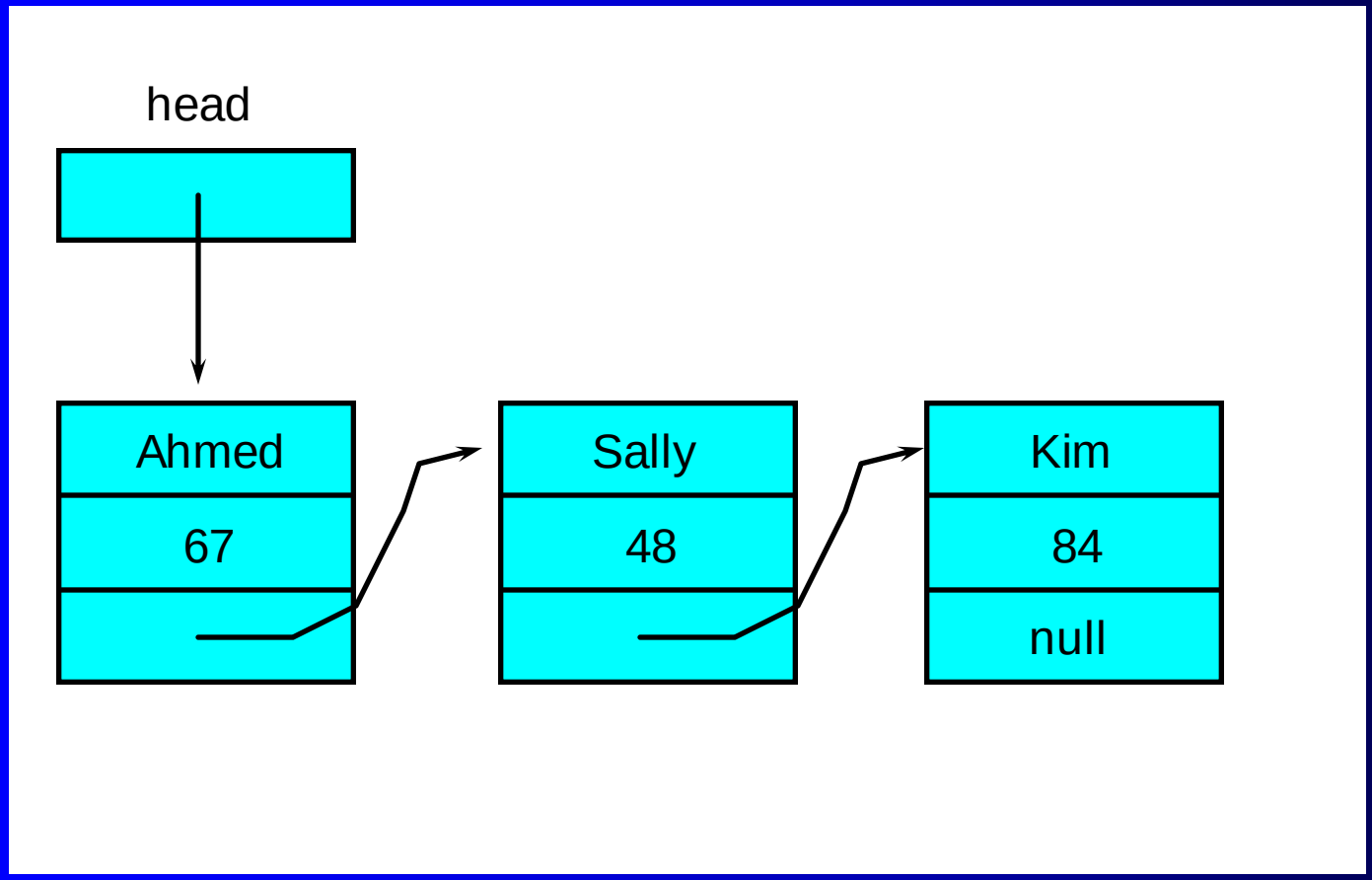
Linked list declaration

Alternatively, the StudentNode class can be defined as an inner class of LinkedList.

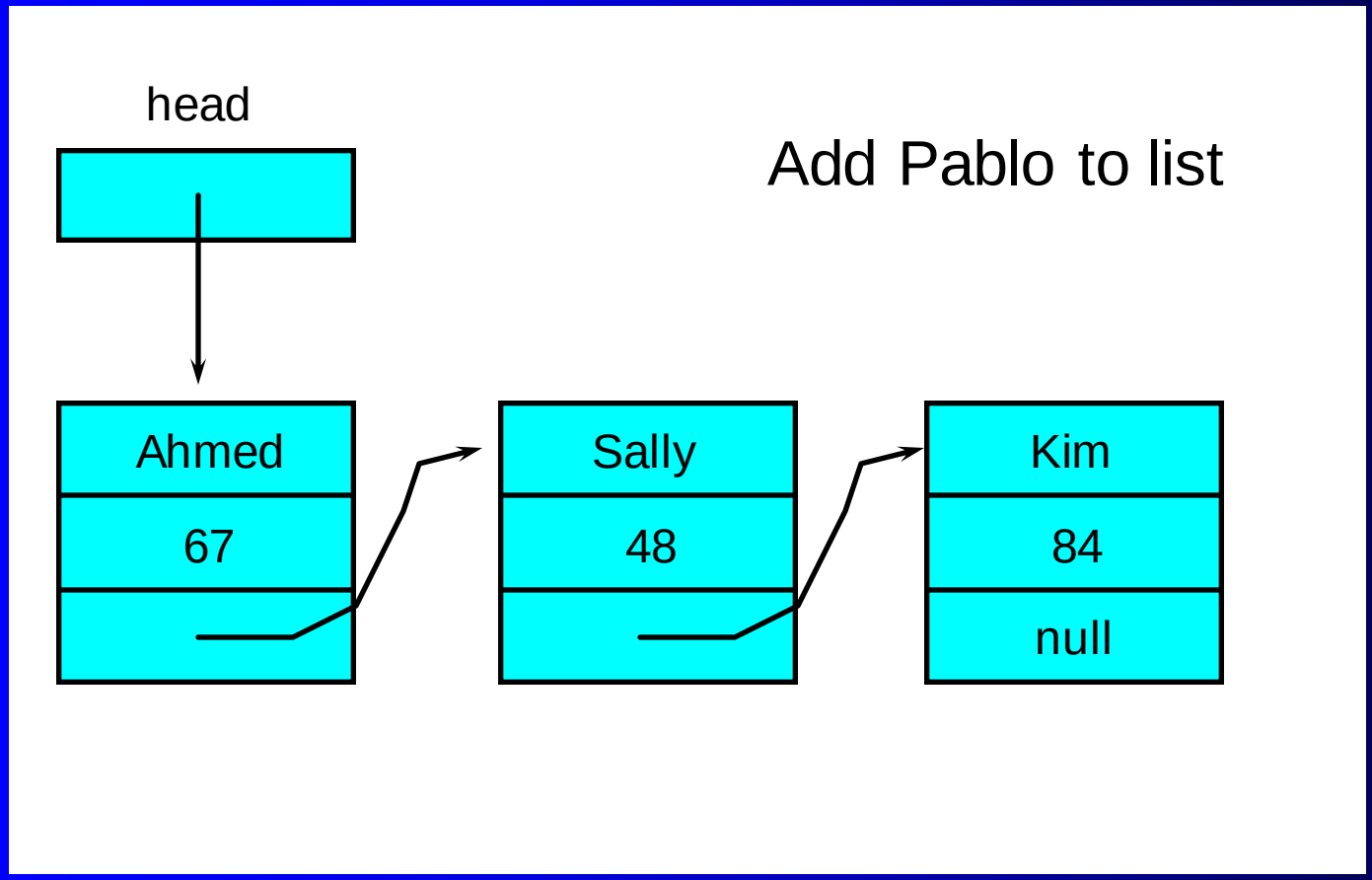
```
class LinkedList {
   private StudentNode head = null;
   ... ...
   private class StudentNode {
            private String name;
            private int mark;
            private StudentNode next;
            public StudentNode(String _n, int _m) {
               name = _n;  mark = _m;
               next = null;
            }
   ... ...
   }
}
```

version 2

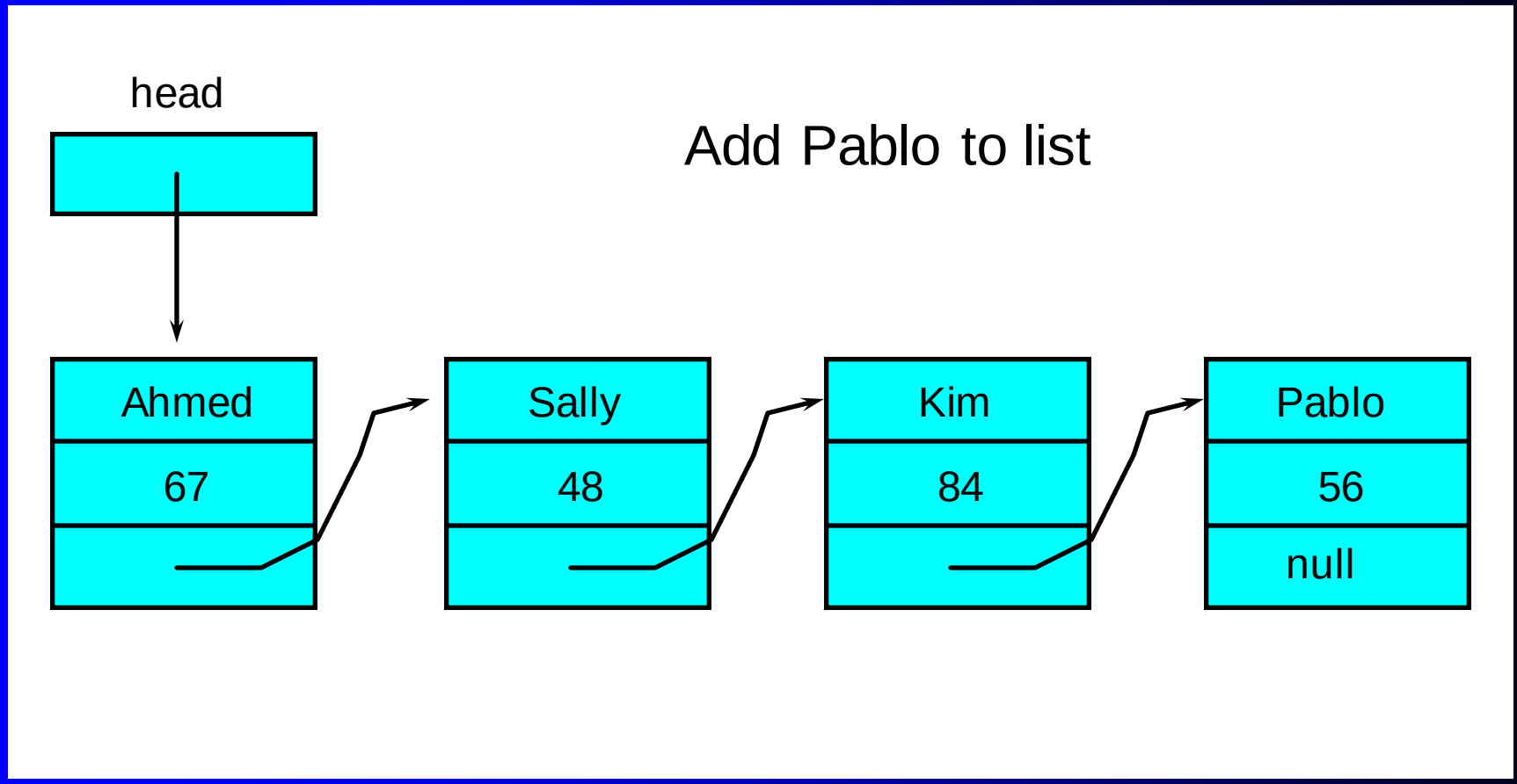
A linked list



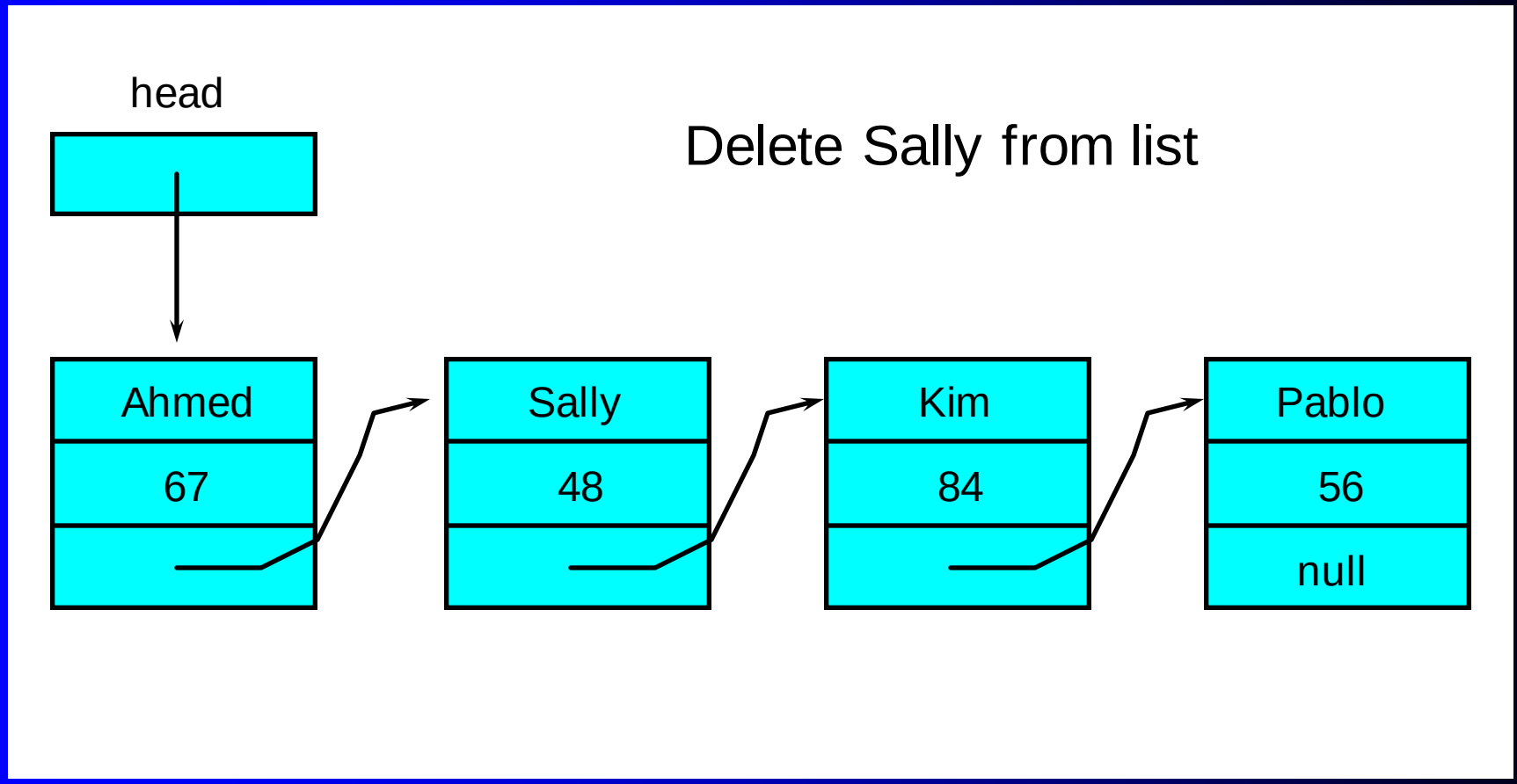
Adding to a linked list



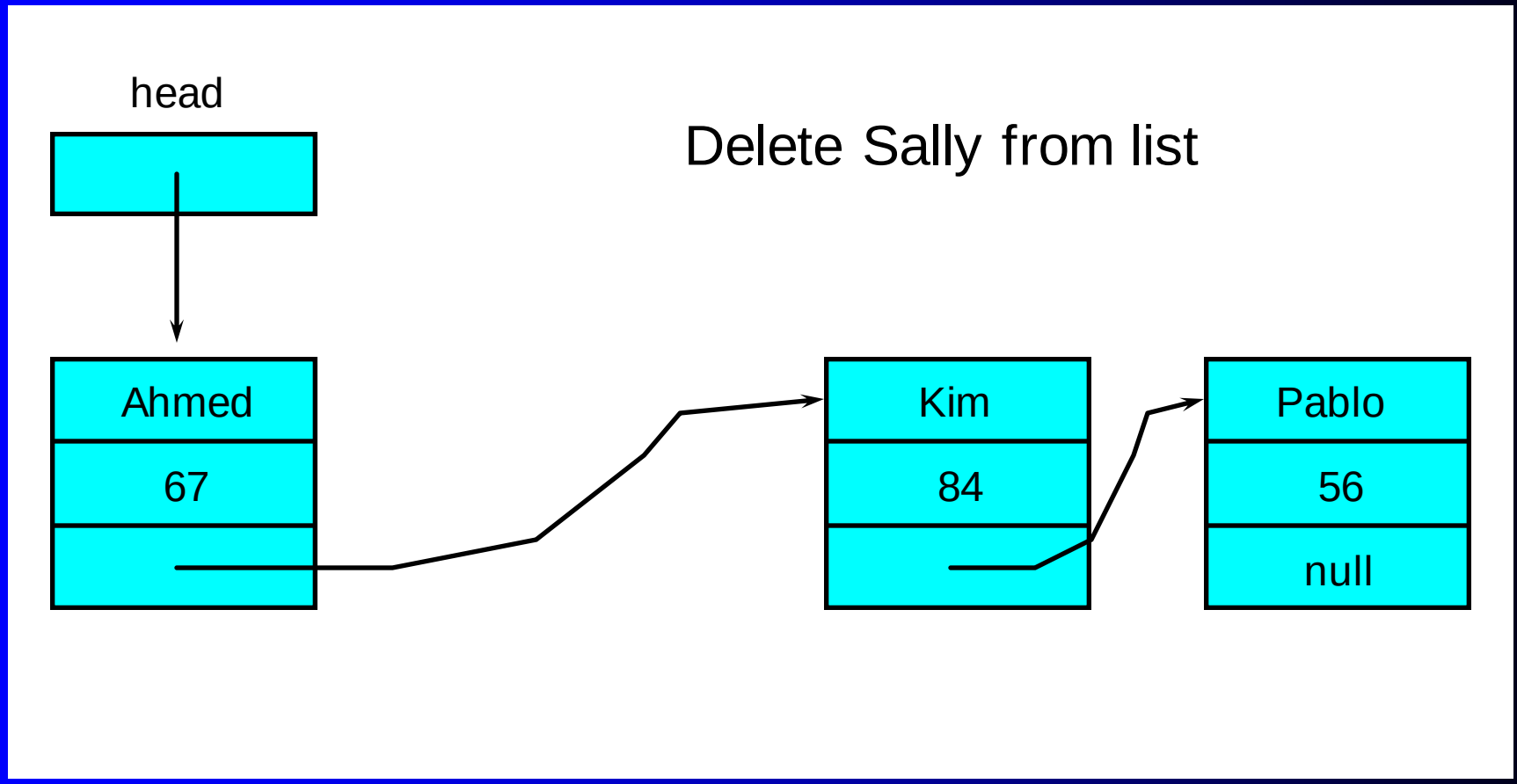
Adding to a linked list



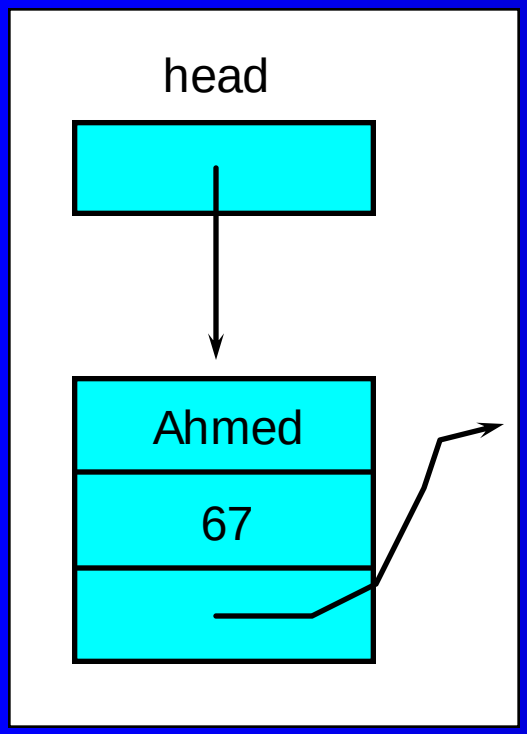
Deleting from a linked list



Deleting from a linked list



Accessing elements



Accessing the first element

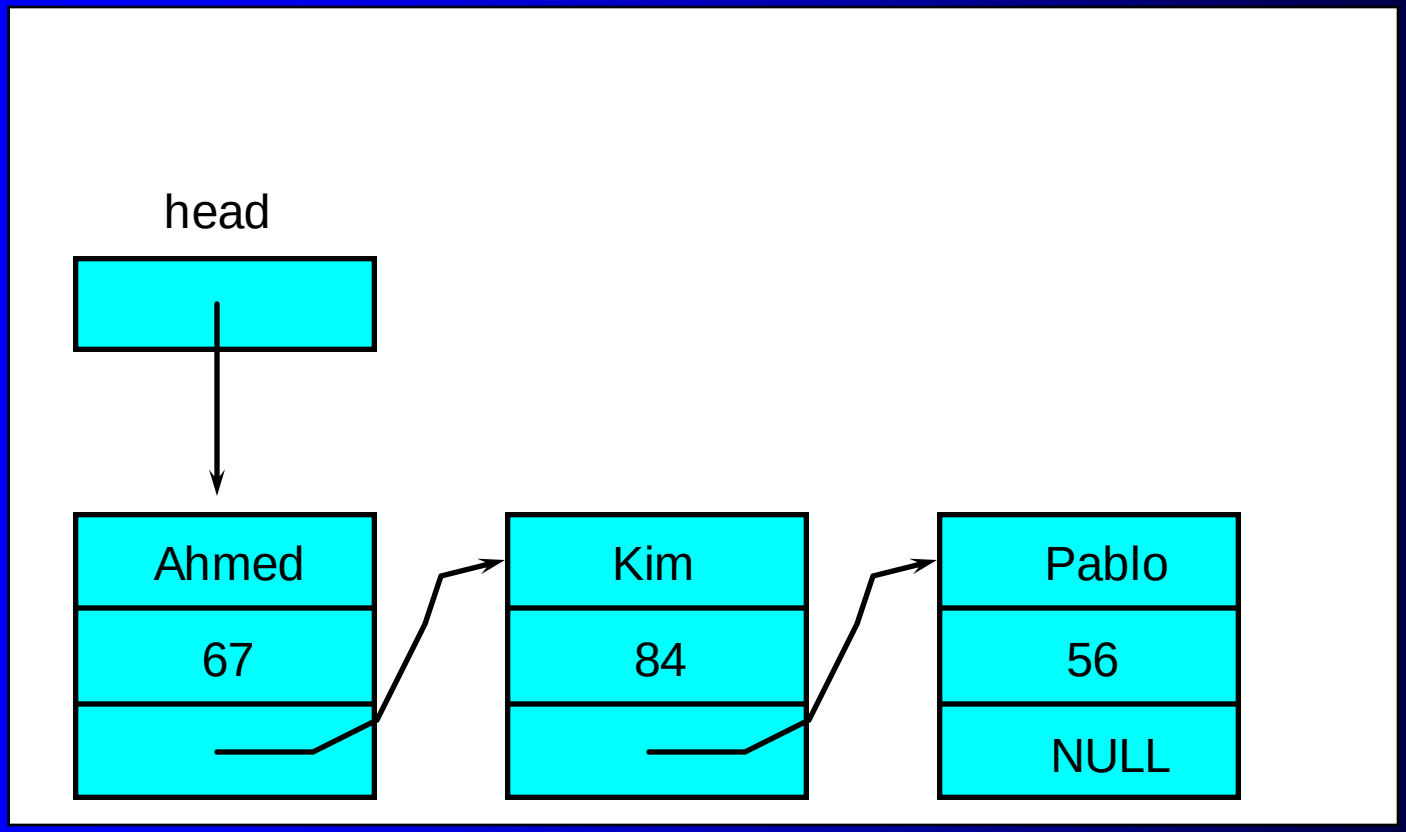
head.mark

head.name

is this using version 1 or 2 ?

Insertion at front of list

- Add a node for student called Janice who has a mark of 89

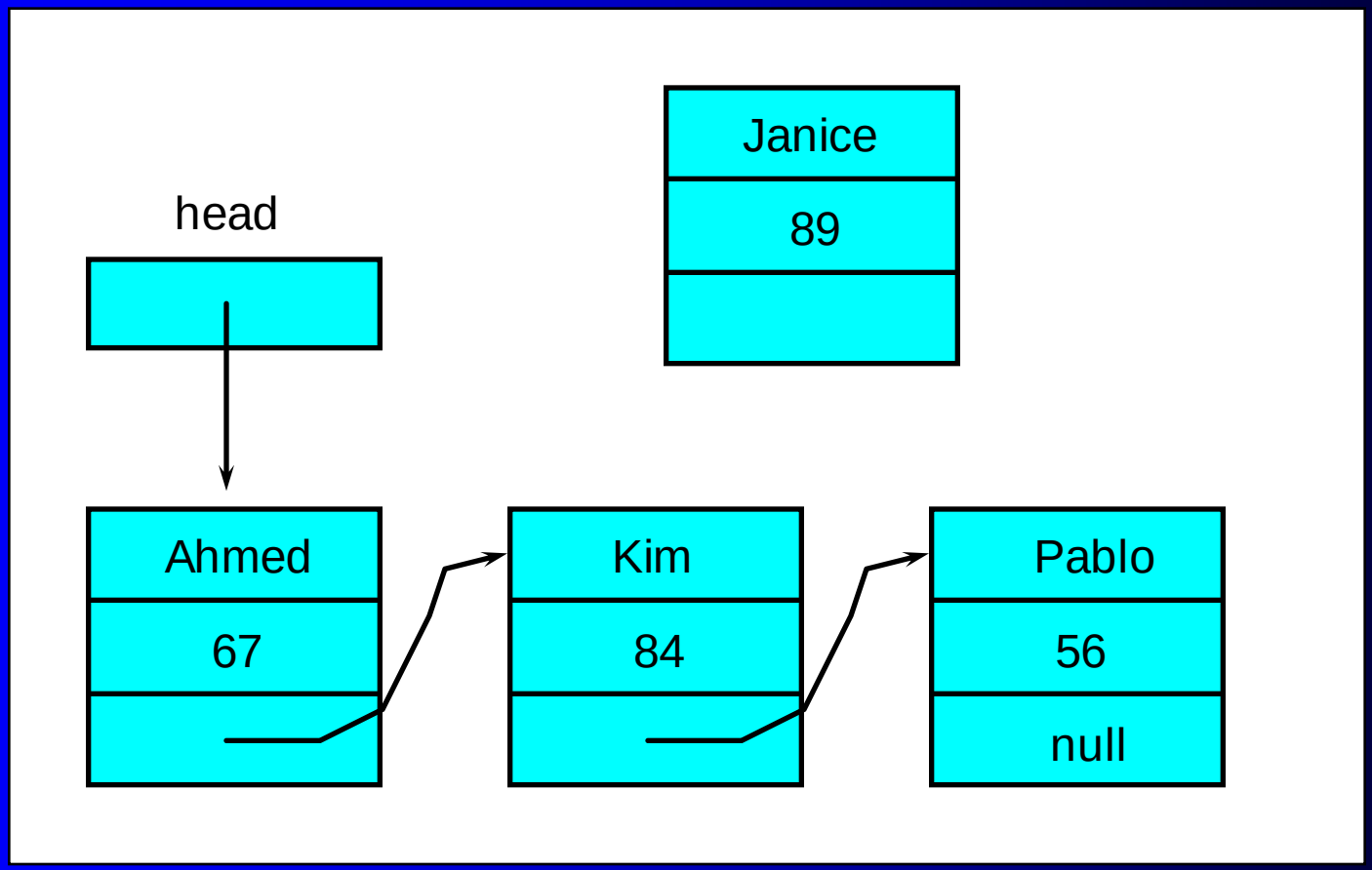


Insertion (ctd)

- Design
  - create a new node
  - fill in its data fields
  - connect the node to the front of the list
  - change head so that it points to the new node

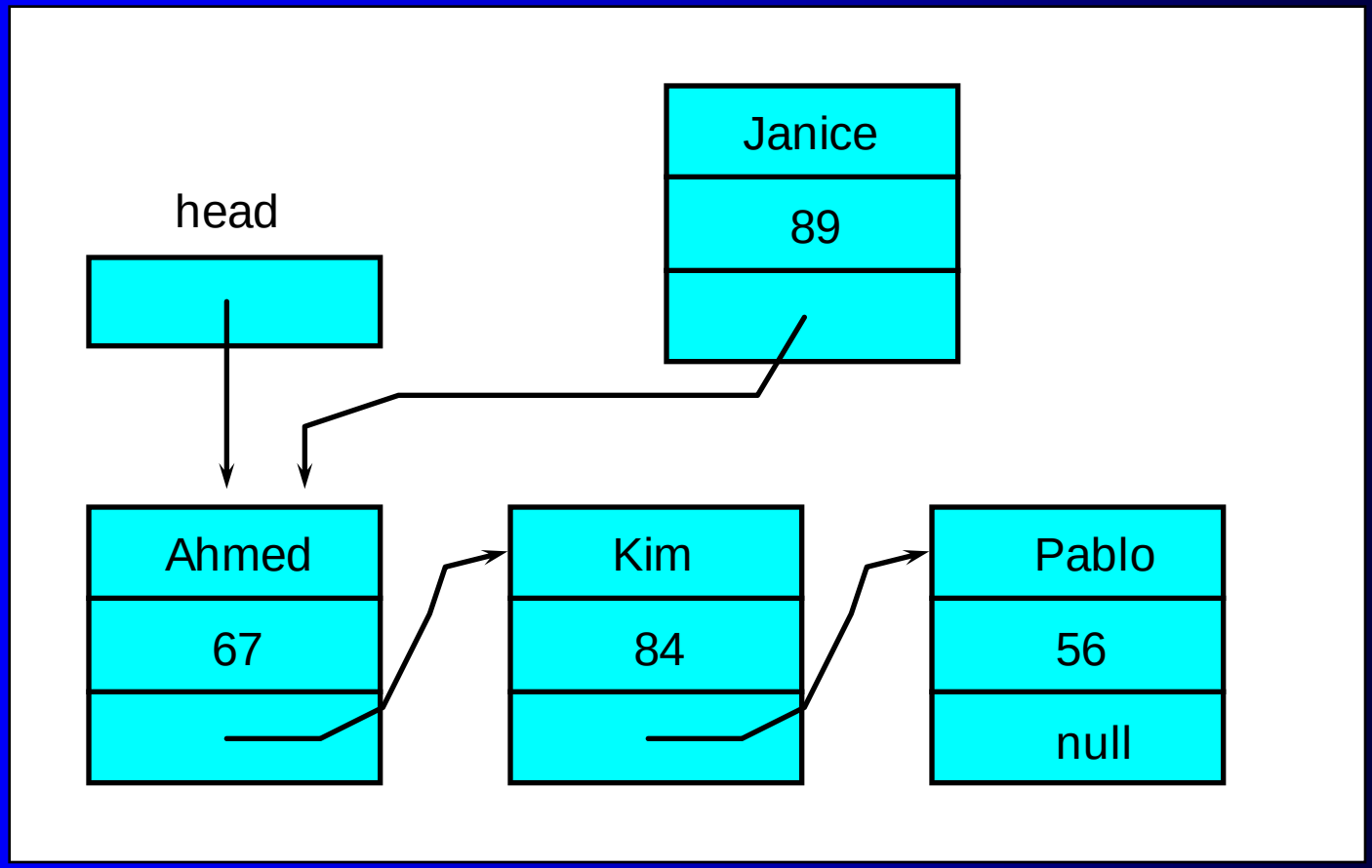
Insertion (ctd)

- Add a node for a student called Janice who has a mark of 89



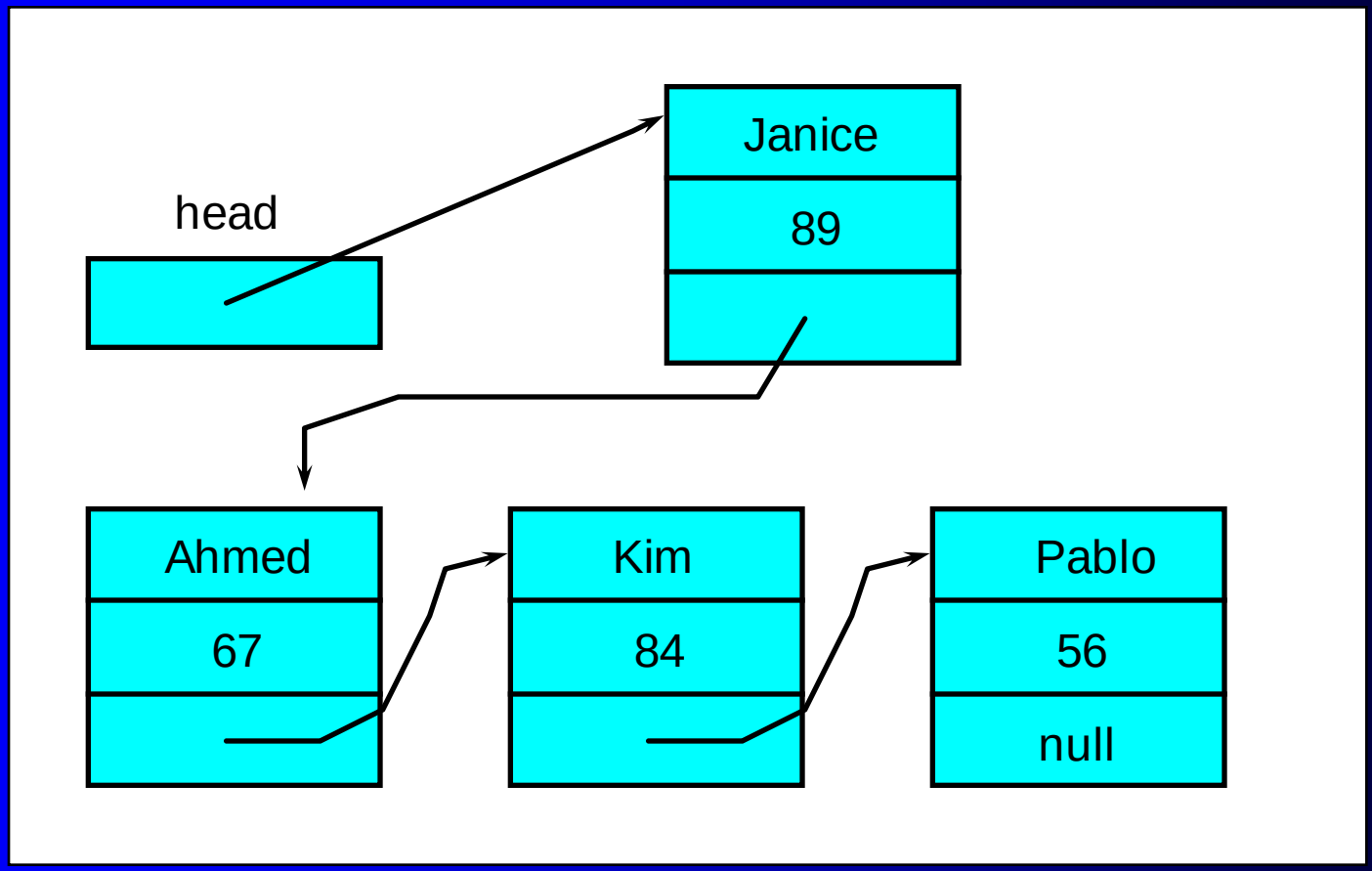
Insertion (ctd)

- Add a node for a student called Janice who has a mark of 89



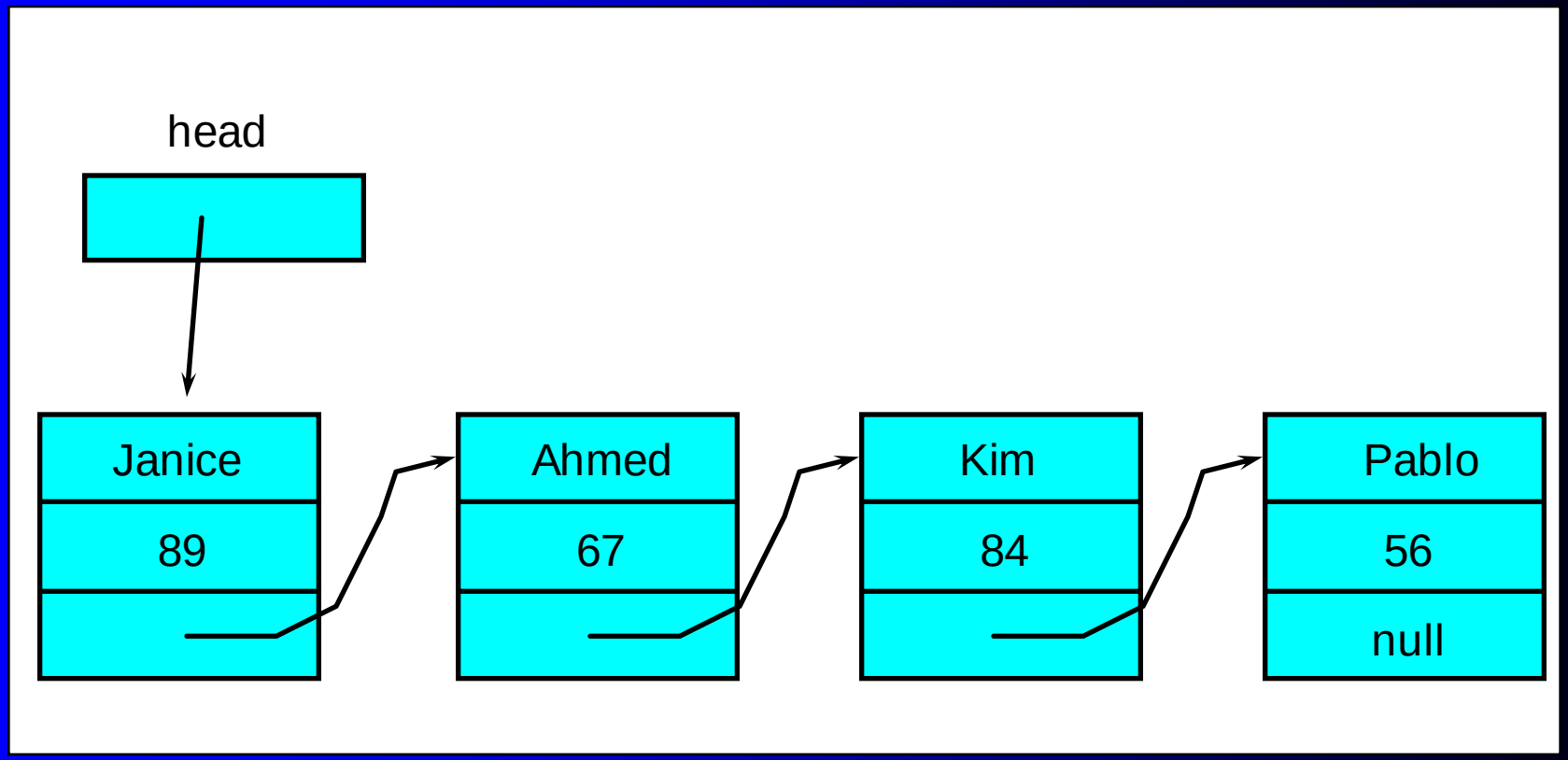
Insertion (ctd)

- Add a node for a student called Janice who has a mark of 89



Insertion (ctd)

- Add a node for a student called Janice who has a mark of 89



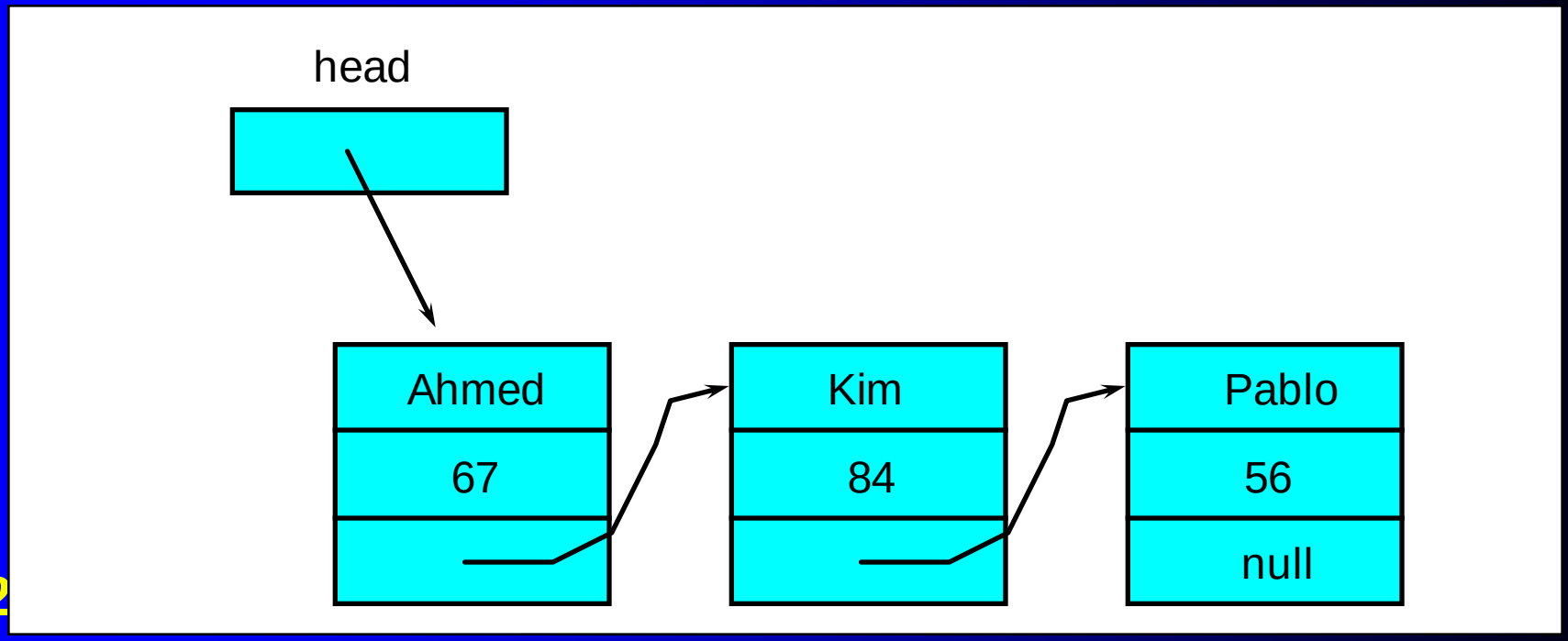
Insertion (ctd)

- JAVA source code

```
StudentNode p = new StudentNode
        (“Janice”, 89);

p.next = head;
head = p;
```

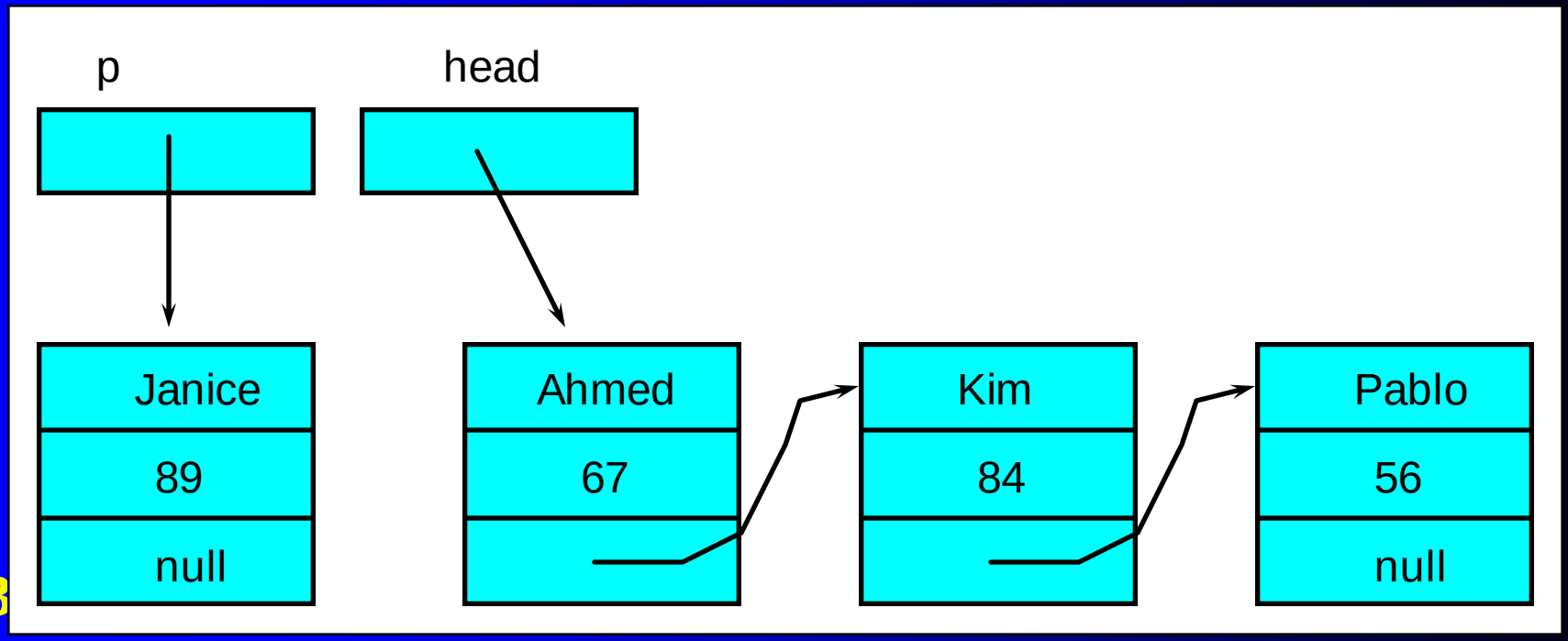
Insertion (ctd)

```
StudentNode p = new StudentNode
                    (“Janice”, 89);

p.next = head;
head = p;
```



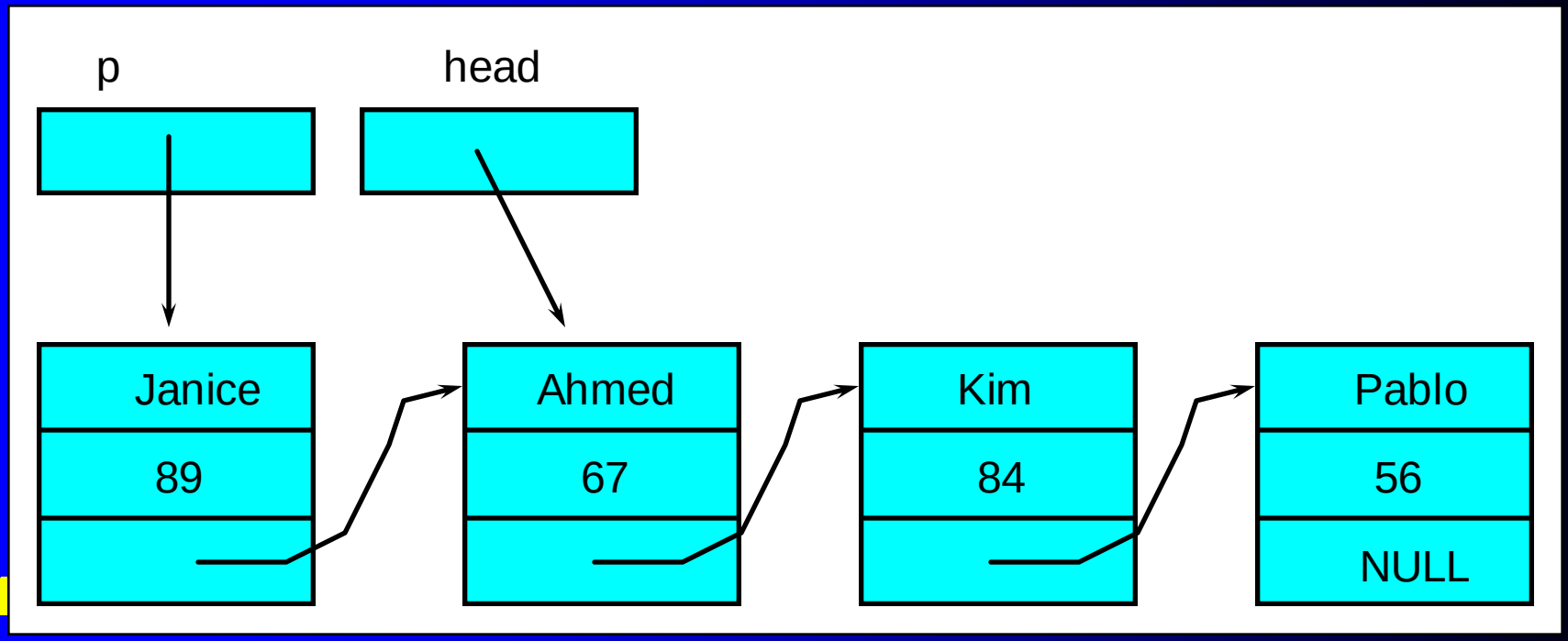
Insertion (ctd)

```
StudentNode p = new StudentNode
        ("Janice", 89);

p.next = head;
head = p;
```



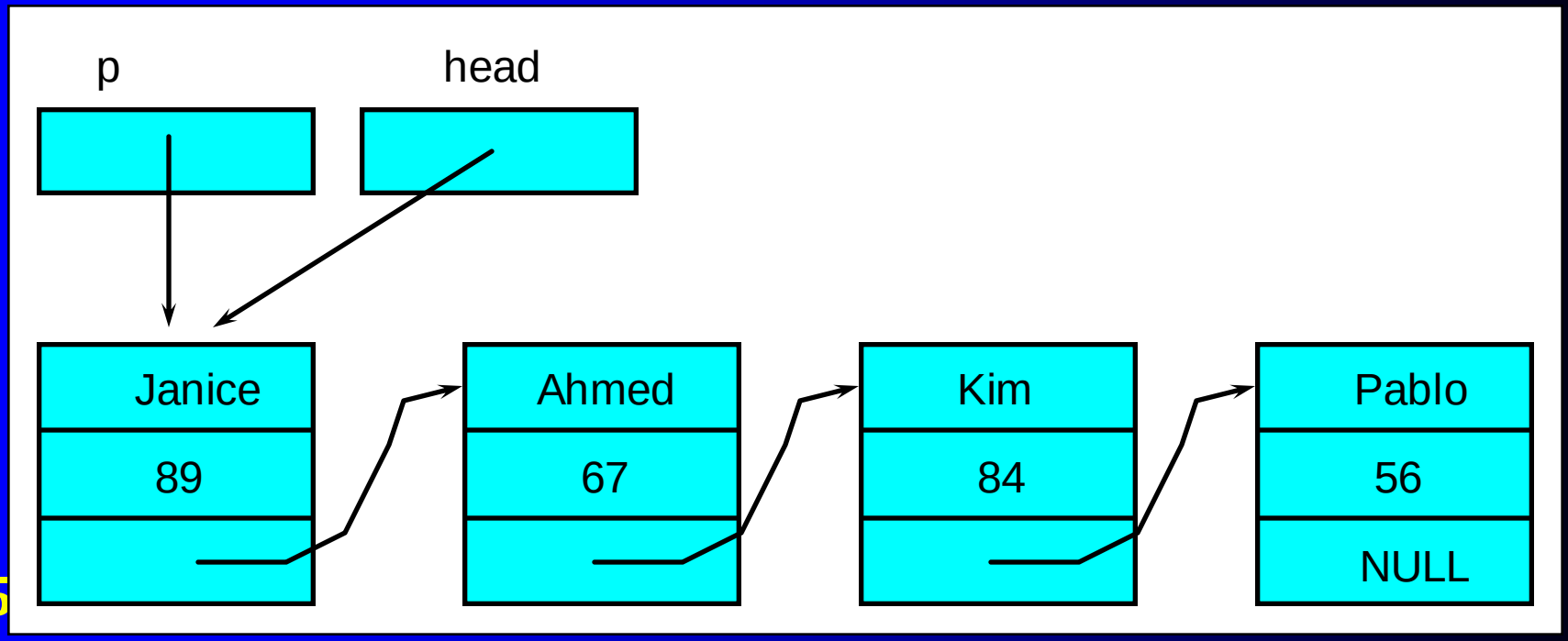
Insertion (ctd)

```
StudentNode p = new StudentNode
            ("Janice", 89);

p.next = head;
head = p;
```



Insertion (ctd)

```
StudentNode p = new StudentNode
            ("Janice", 89);

p.next = head;
head = p;
```

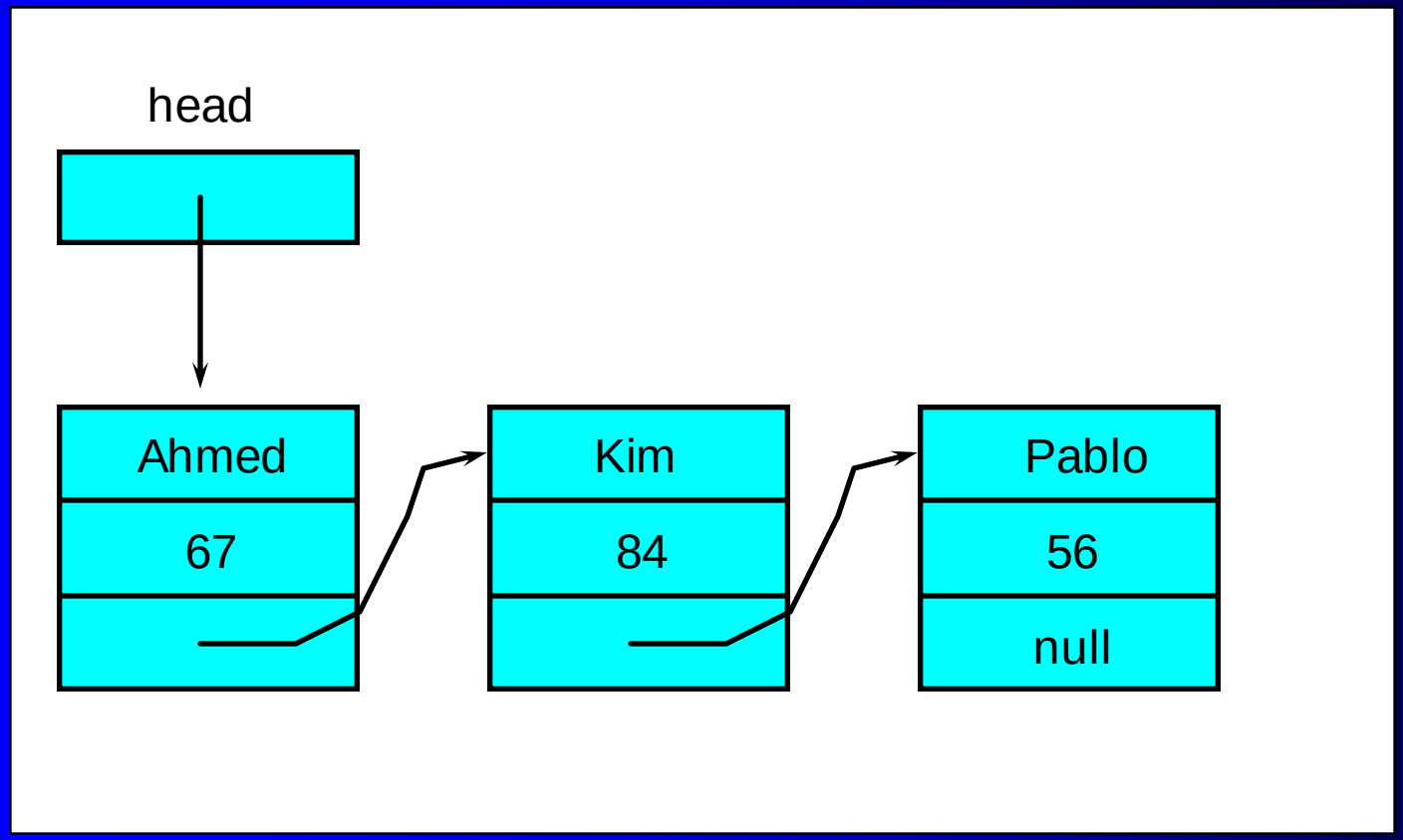


Building a linked list

- To create a linked list:

```
head = null
while more nodes
   insert a node at front of list
```

Printing a linked list



Printing a linked list (ctd)

- Pseudocode

start at the head of the list

while there is a node
    print data of the node
    move to the next node on the list

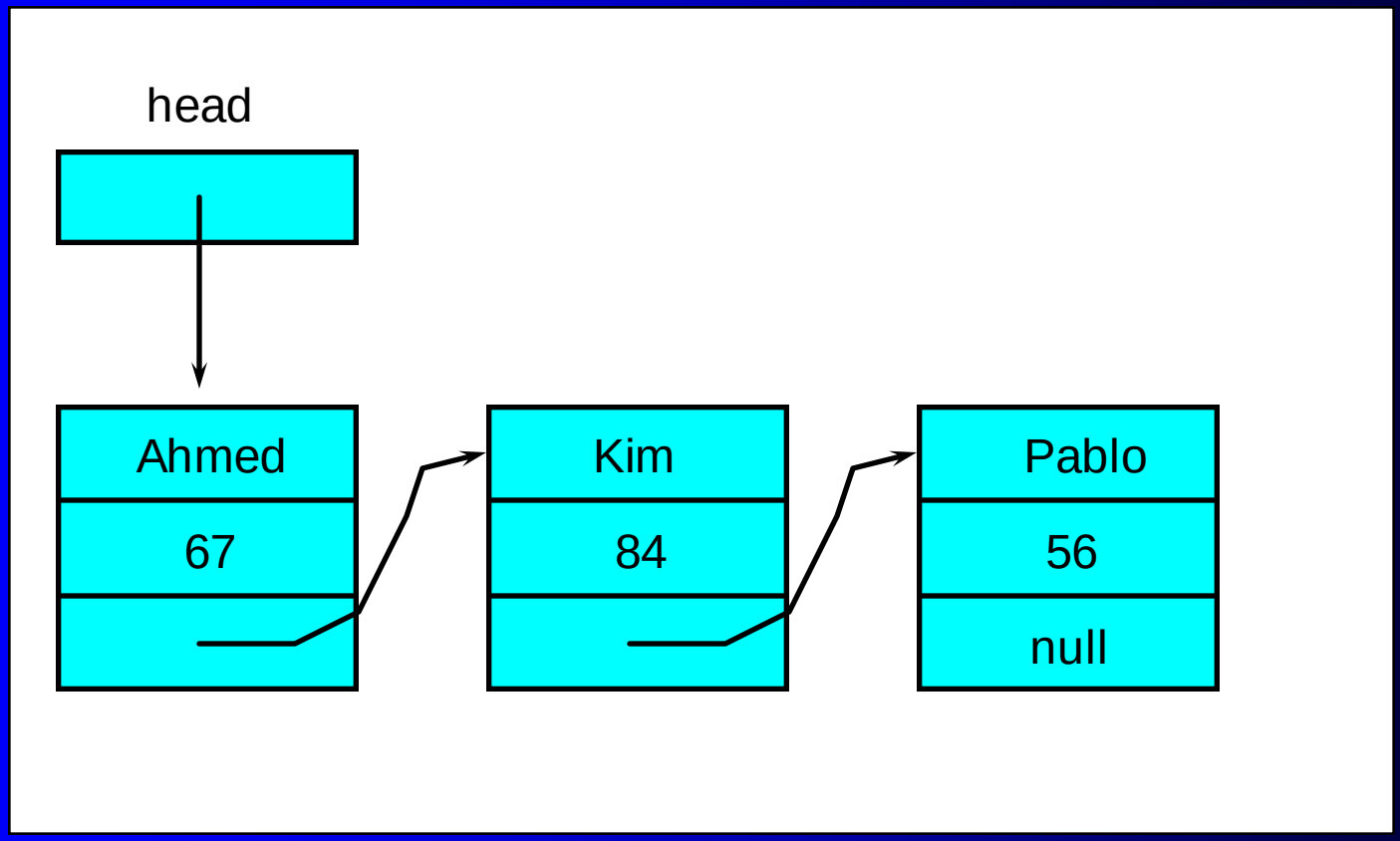
Printing a linked list (ctd)

- JAVA source code

```
StudentNode p = head;

while (p != null) {
   System.out.println (p.name + ": " + p.mark);
   p = p.next;
}
```

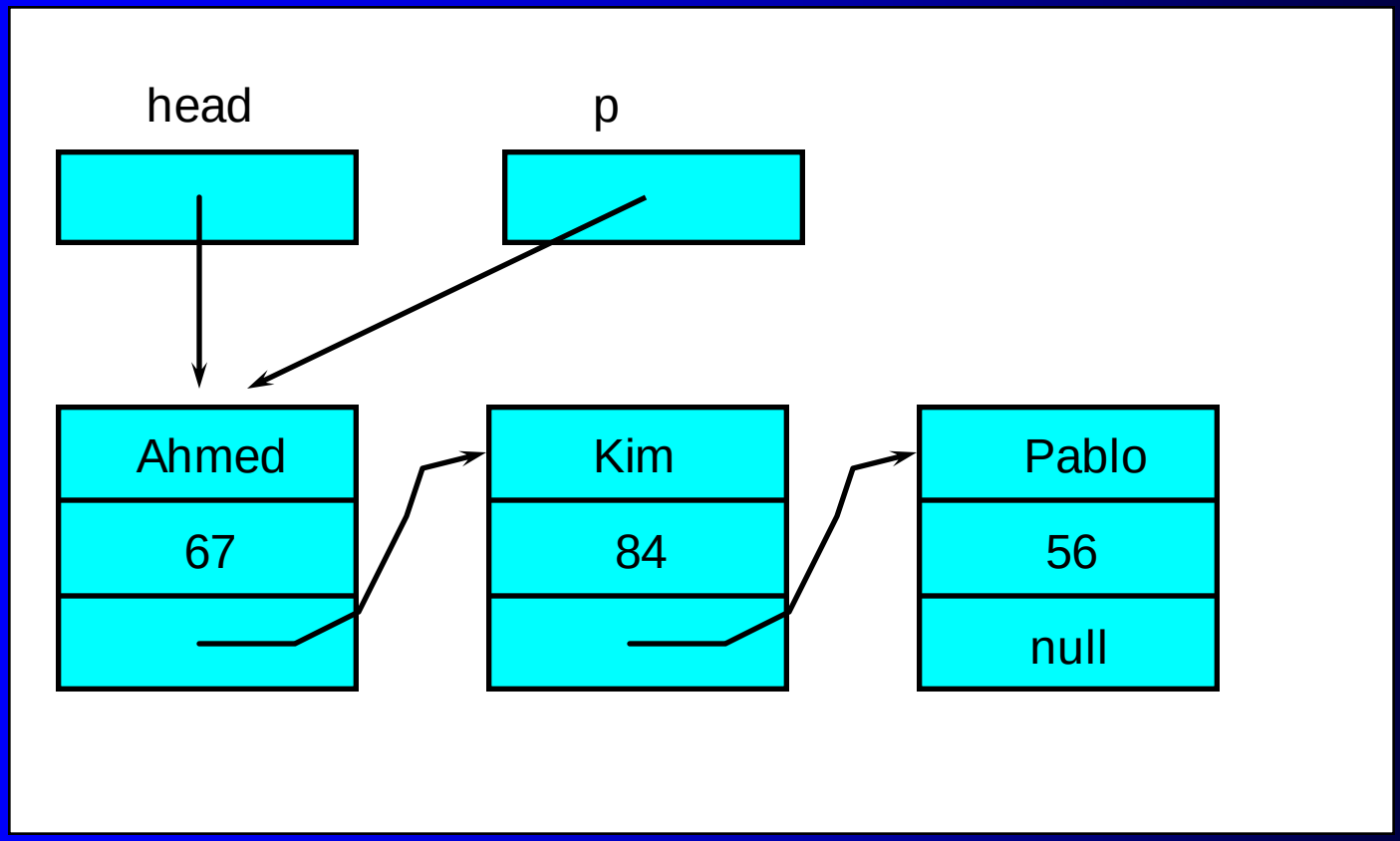
Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + ": " +
p.mark);
   p = p.next;
}
```



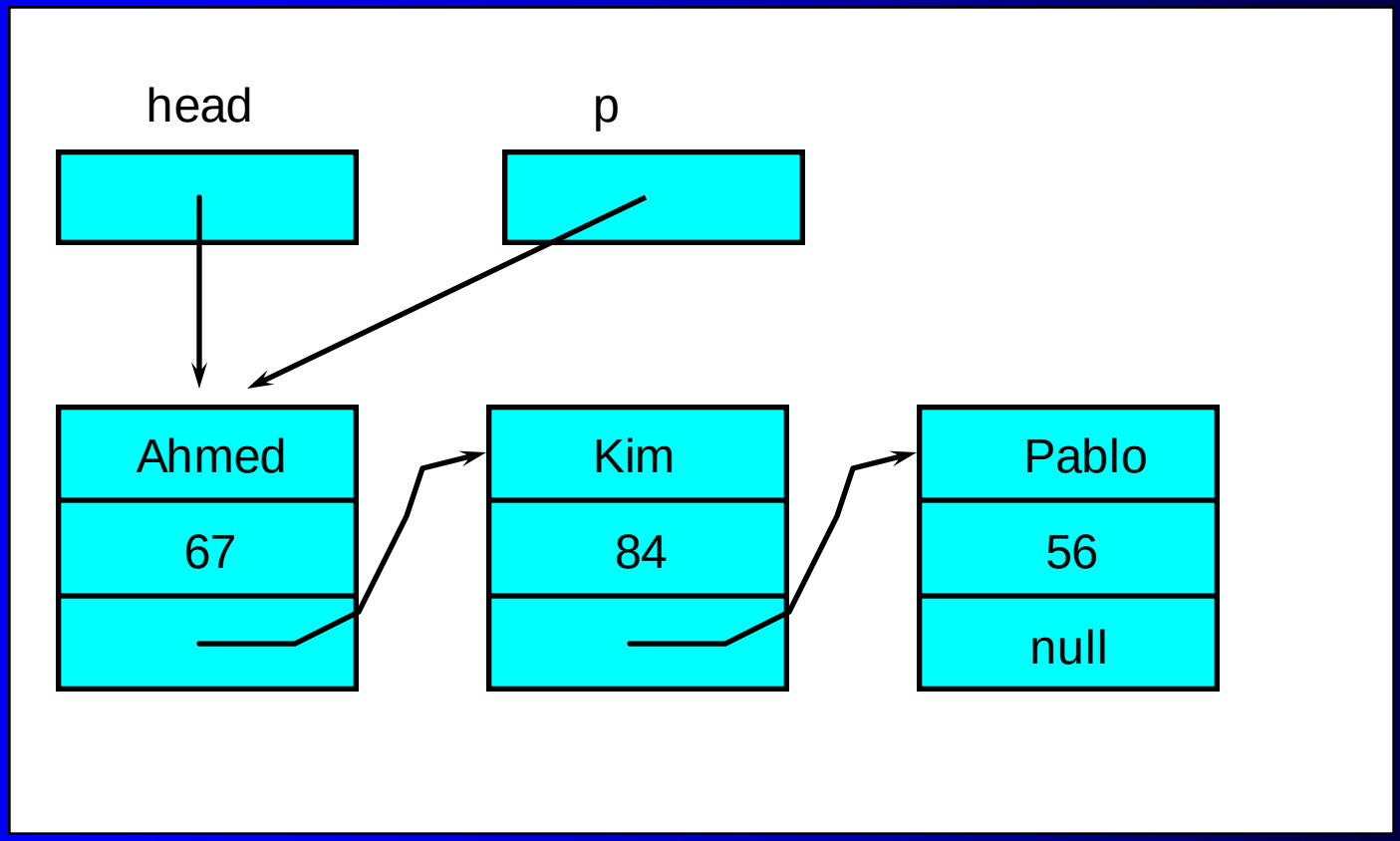
Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + ": " +
p.mark);
   p = p.next;
}
```



Printing a linked list (ctd)

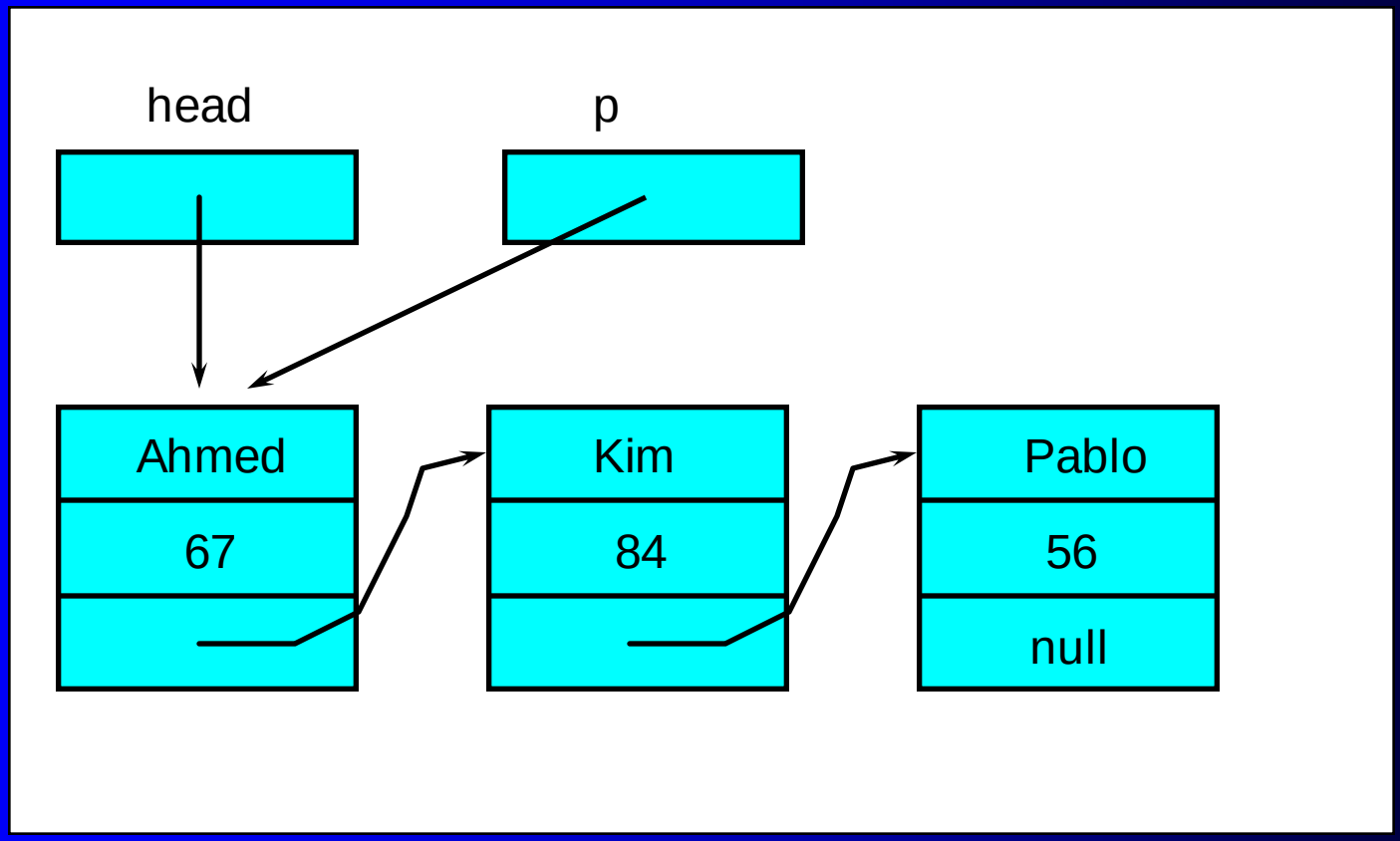
```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + ": " +
p.mark);
   p = p.next;
}
```



Printing a linked list (ctd)

Ahmed: 67

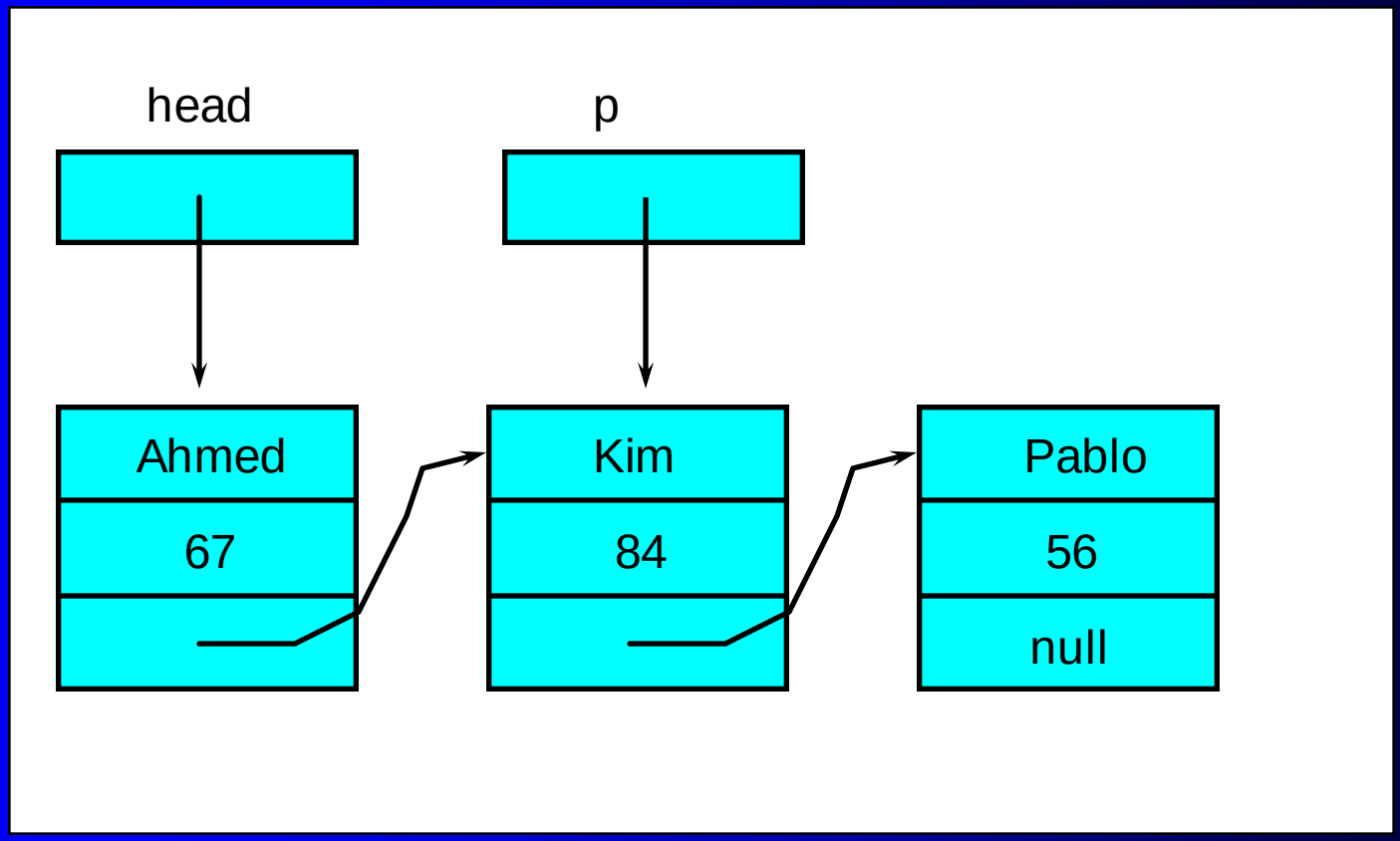
```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + ": " +
p.mark);
   p = p.next;
}
```



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

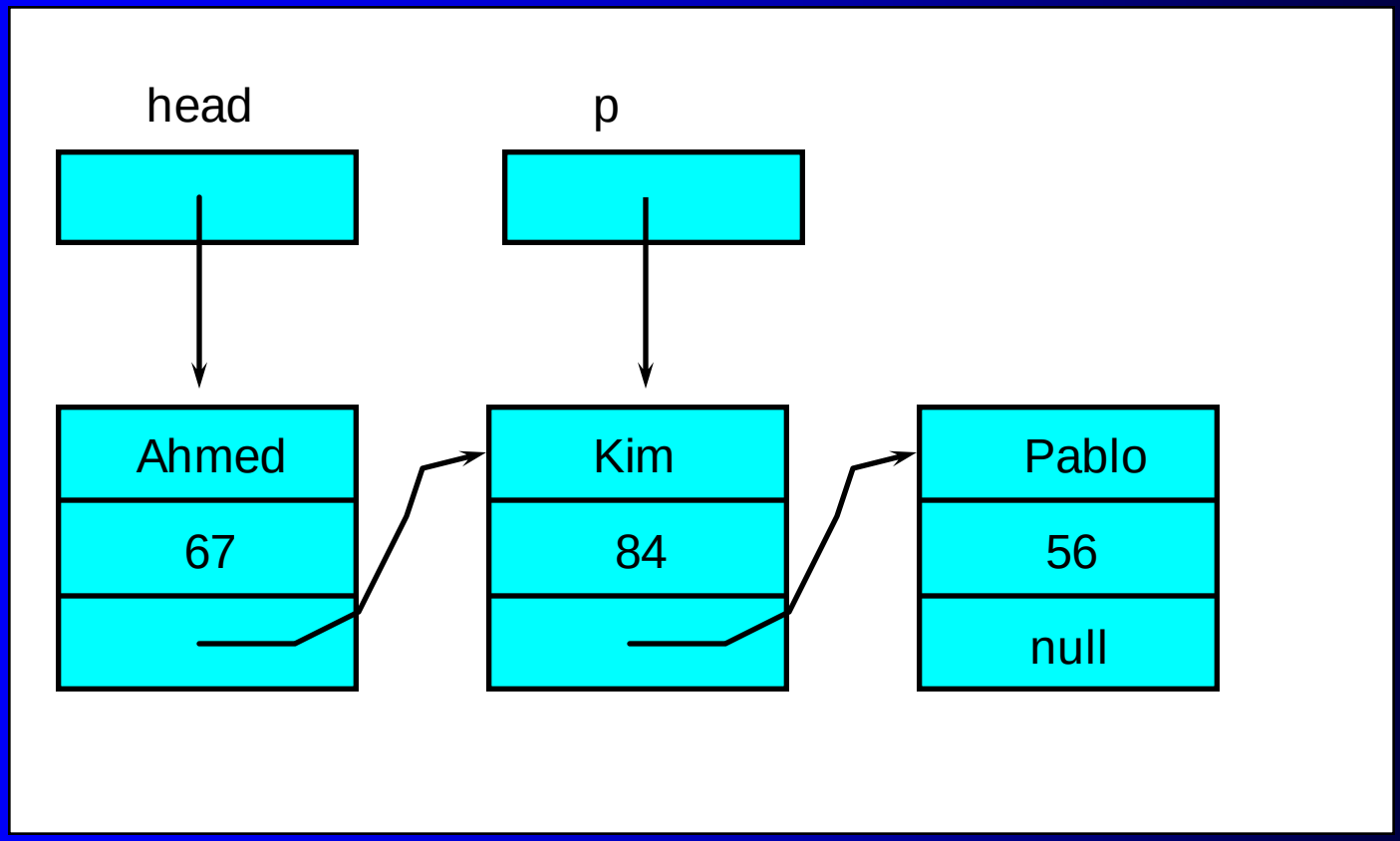
Ahmed: 67



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

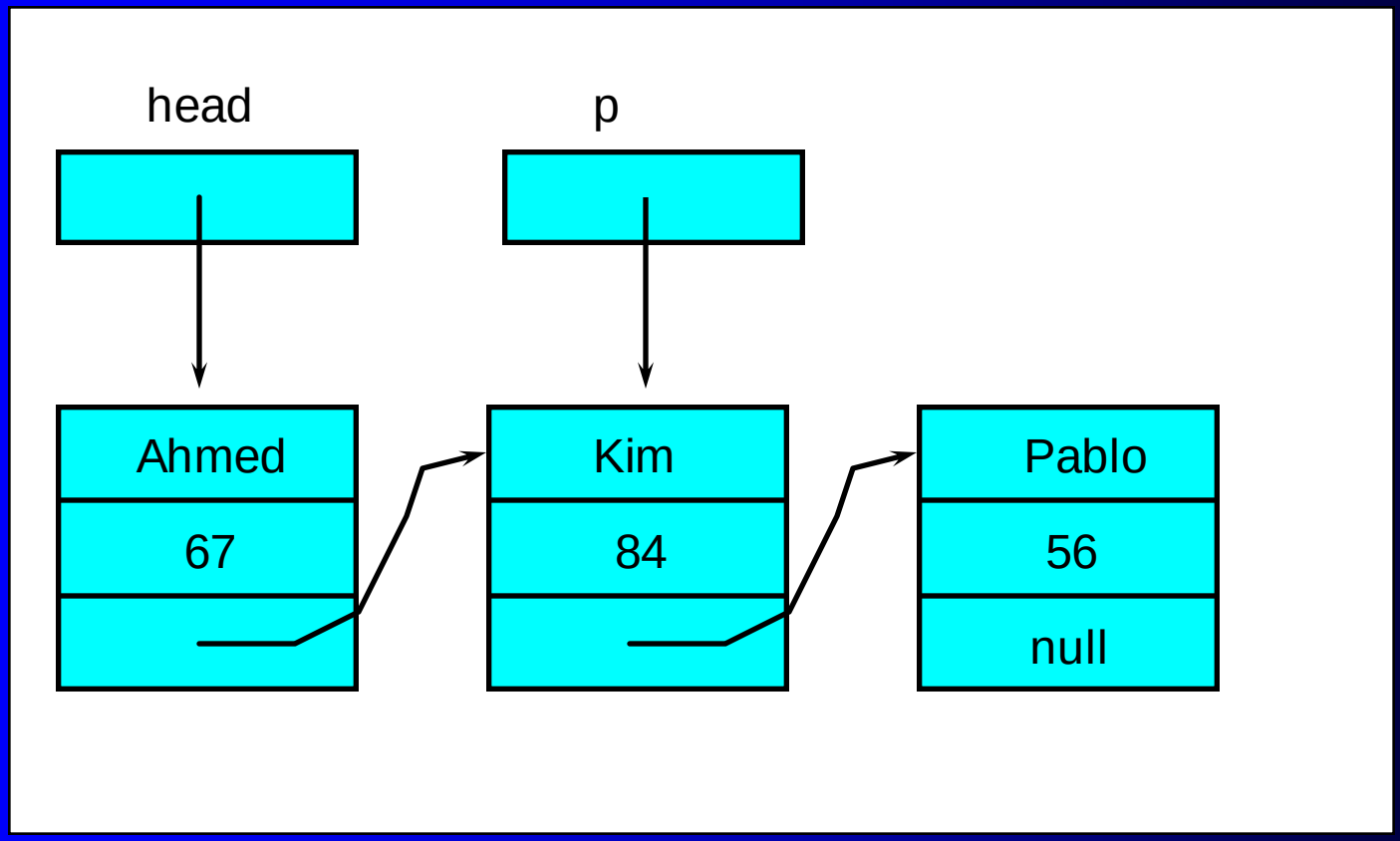
Ahmed: 67



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

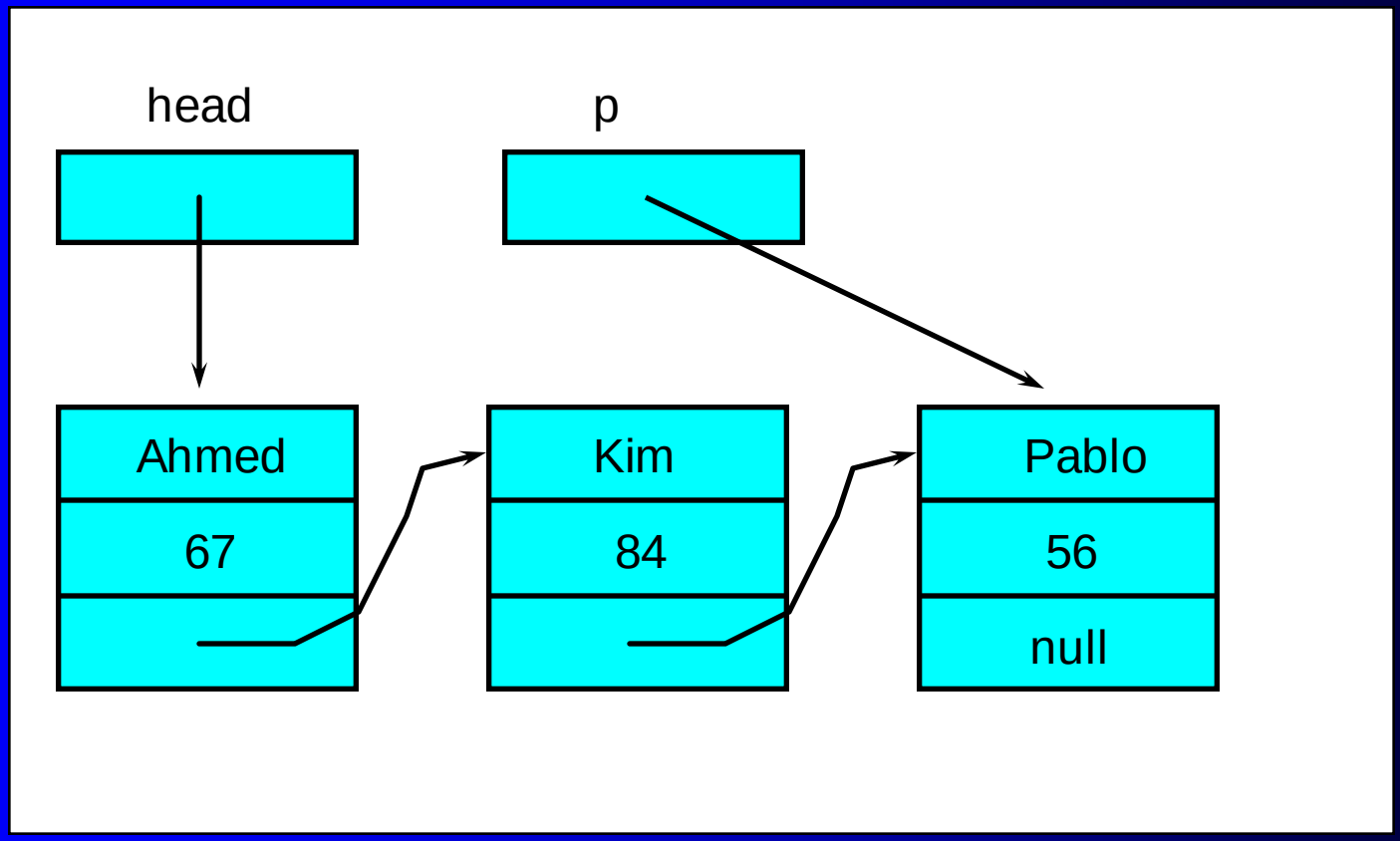
Ahmed: 67
Kim: 84



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

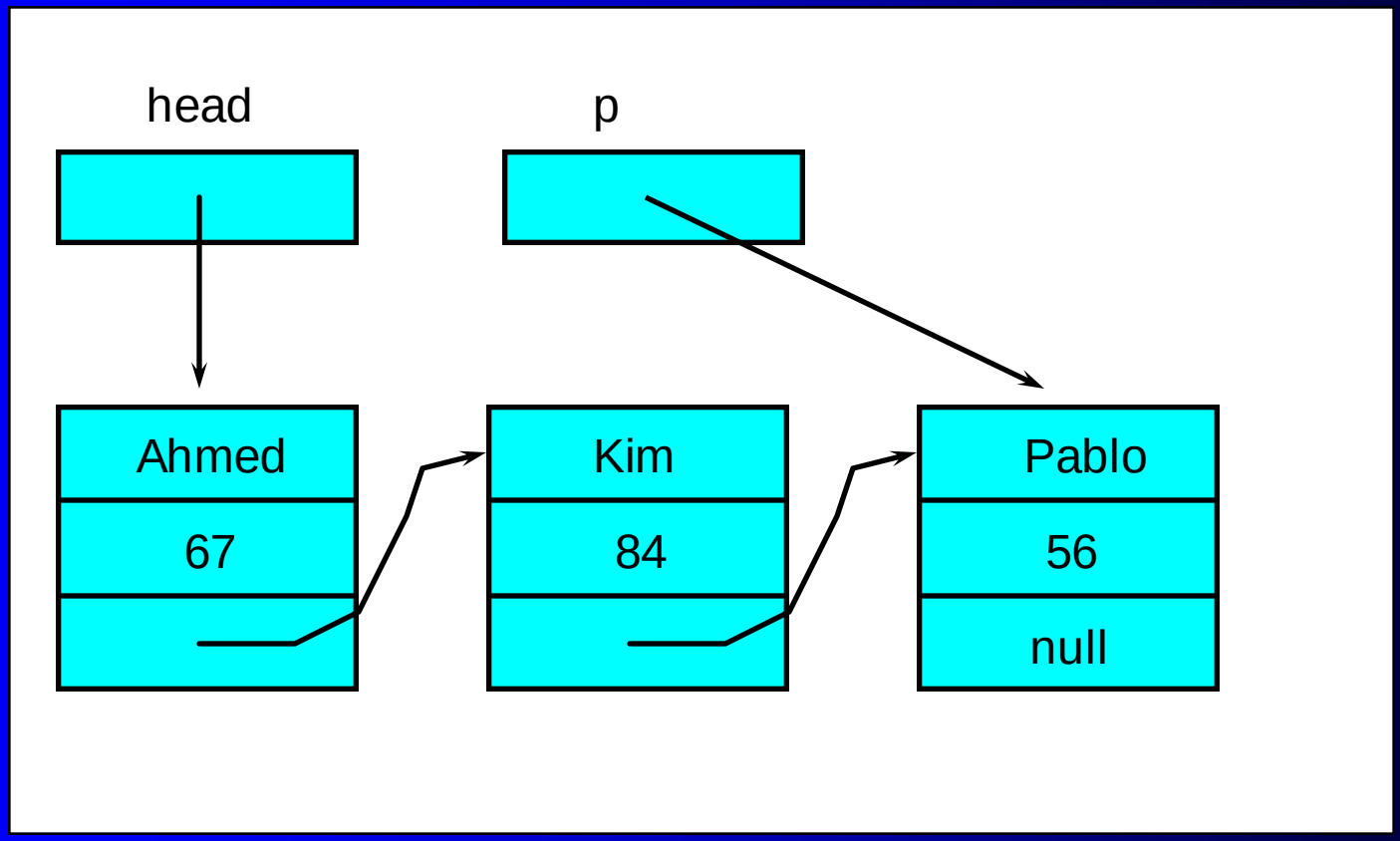
Ahmed: 67
Kim: 84



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

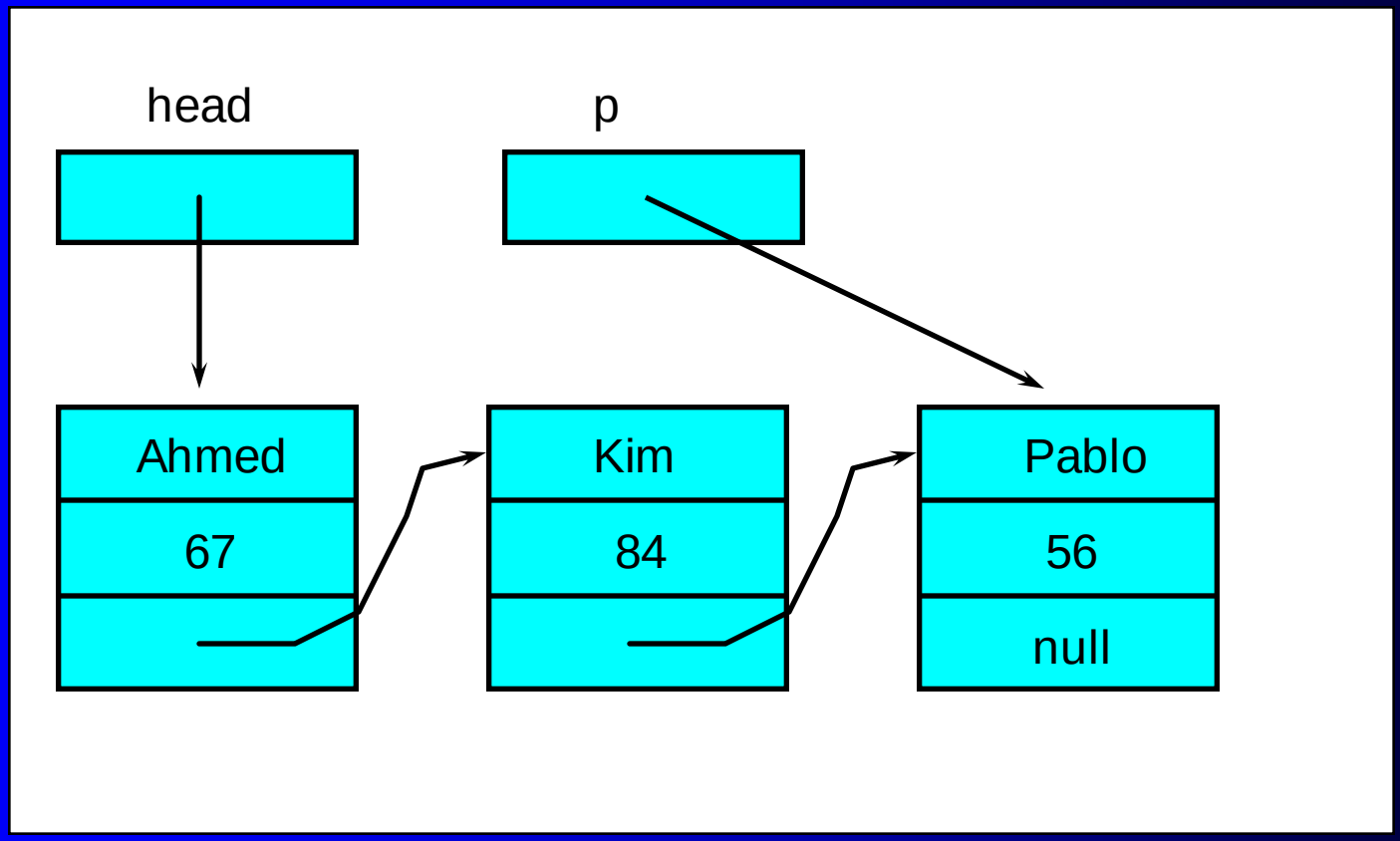
Ahmed: 67
Kim: 84



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

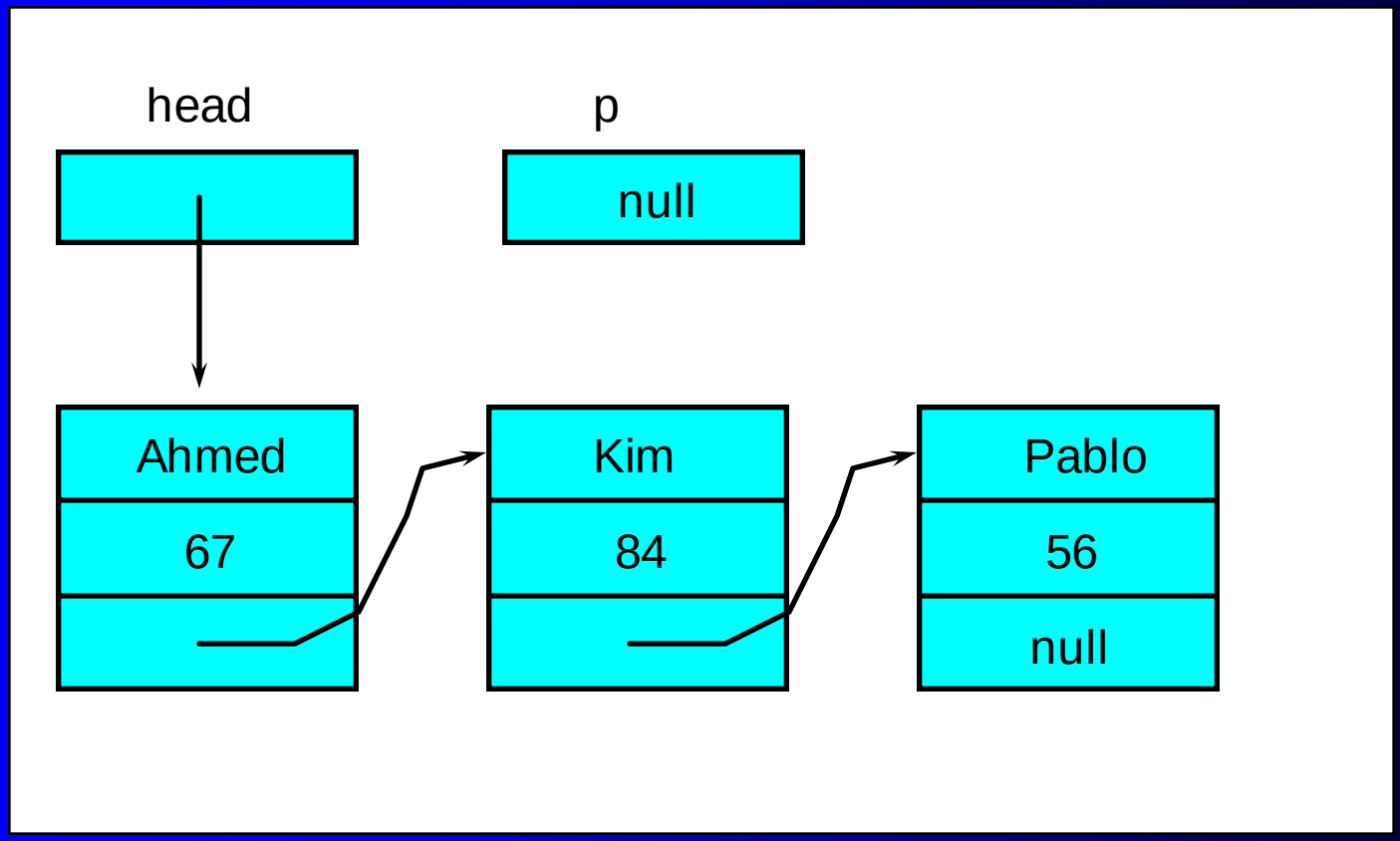
Ahmed: 67
Kim: 84
Pablo: 56



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

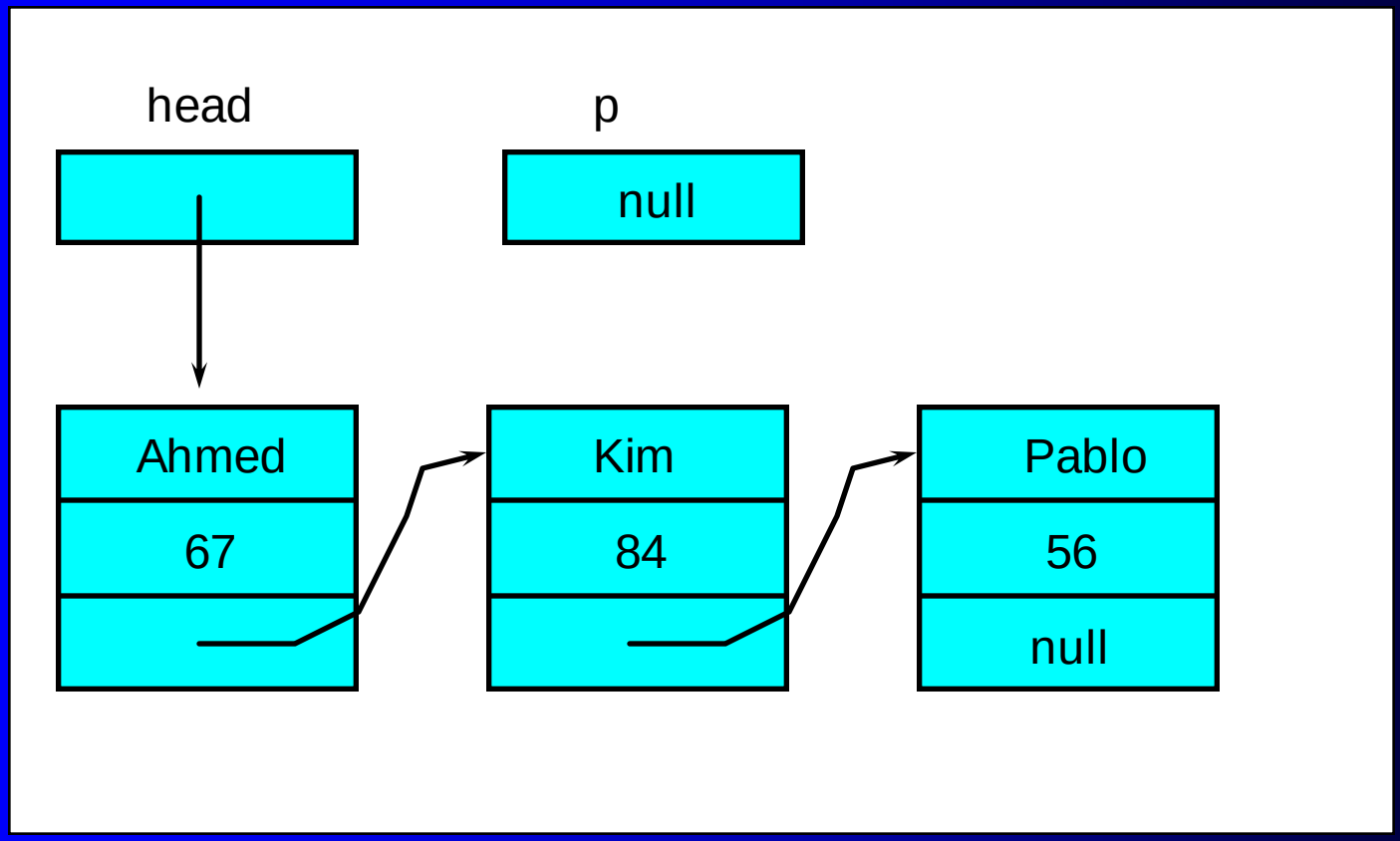
Ahmed: 67
Kim: 84
Pablo: 56



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

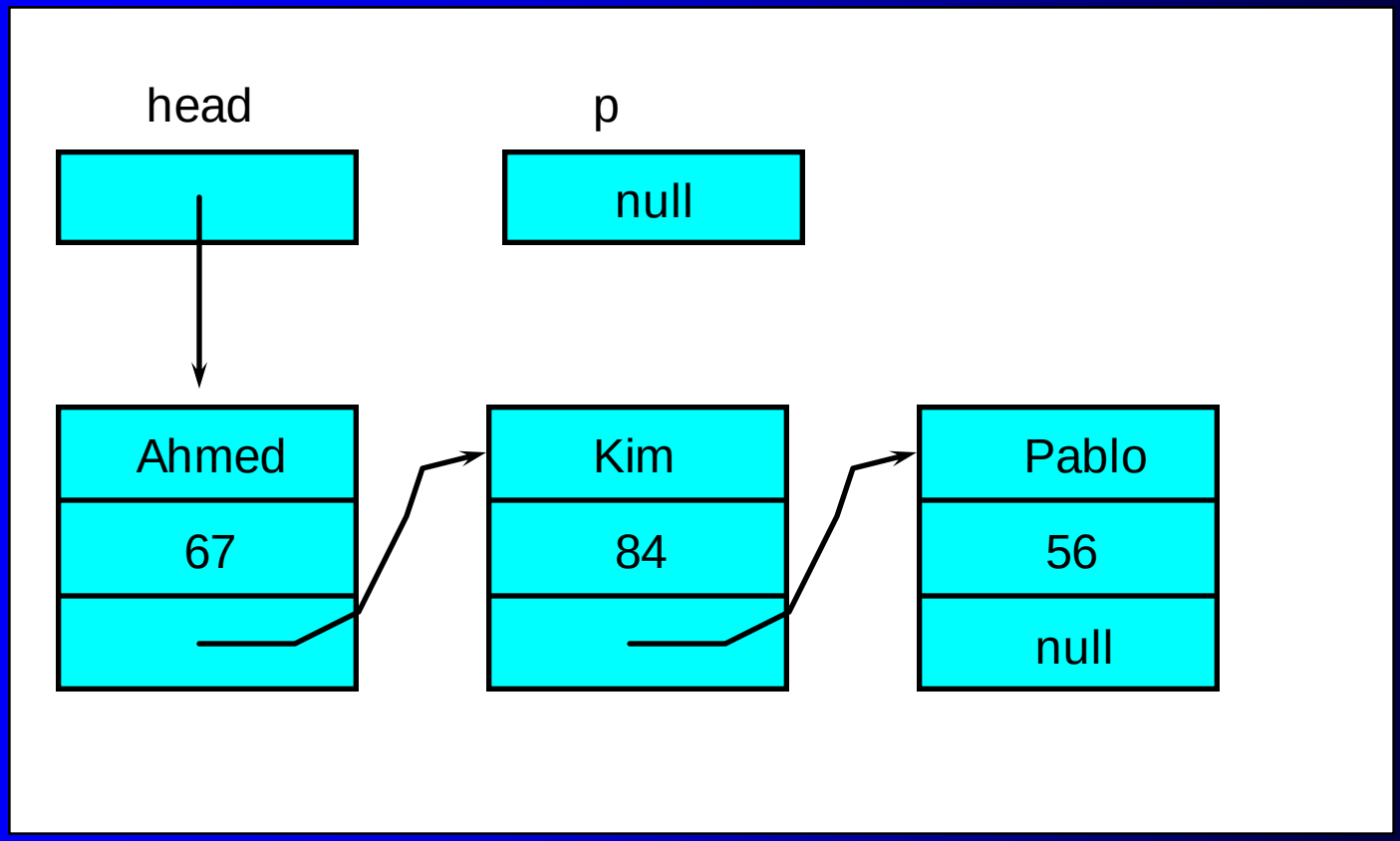
Ahmed: 67
Kim: 84
Pablo: 56



Printing a linked list (ctd)

```
StudentNode p = head;

while (p != null)
{
   System.out.println (p.name + " : "
+ p.mark);
   p = p.next;
}
```

Ahmed: 67
Kim: 84
Pablo: 56



Linked list example

- Problem
  - write pseudocode to work out the average mark of the students in the linked list
  - convert the pseudocode to JAVA code

Calculate the average mark

```
StudentNode p = head;
int avMark = 0, numberSt = 0;

while (p != null) {
   System.out.println (p.name + ": " + p.mark);
   avMark = avMark + _____;
   _____;
   p = p.next;
}

if (numberSt == 0)
   _____;___
else
   _____;
```