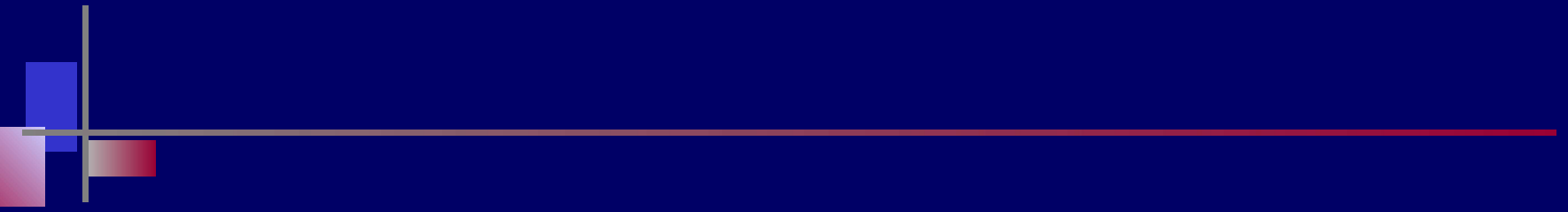


# CSCP-2034: Data Structures and Algorithms

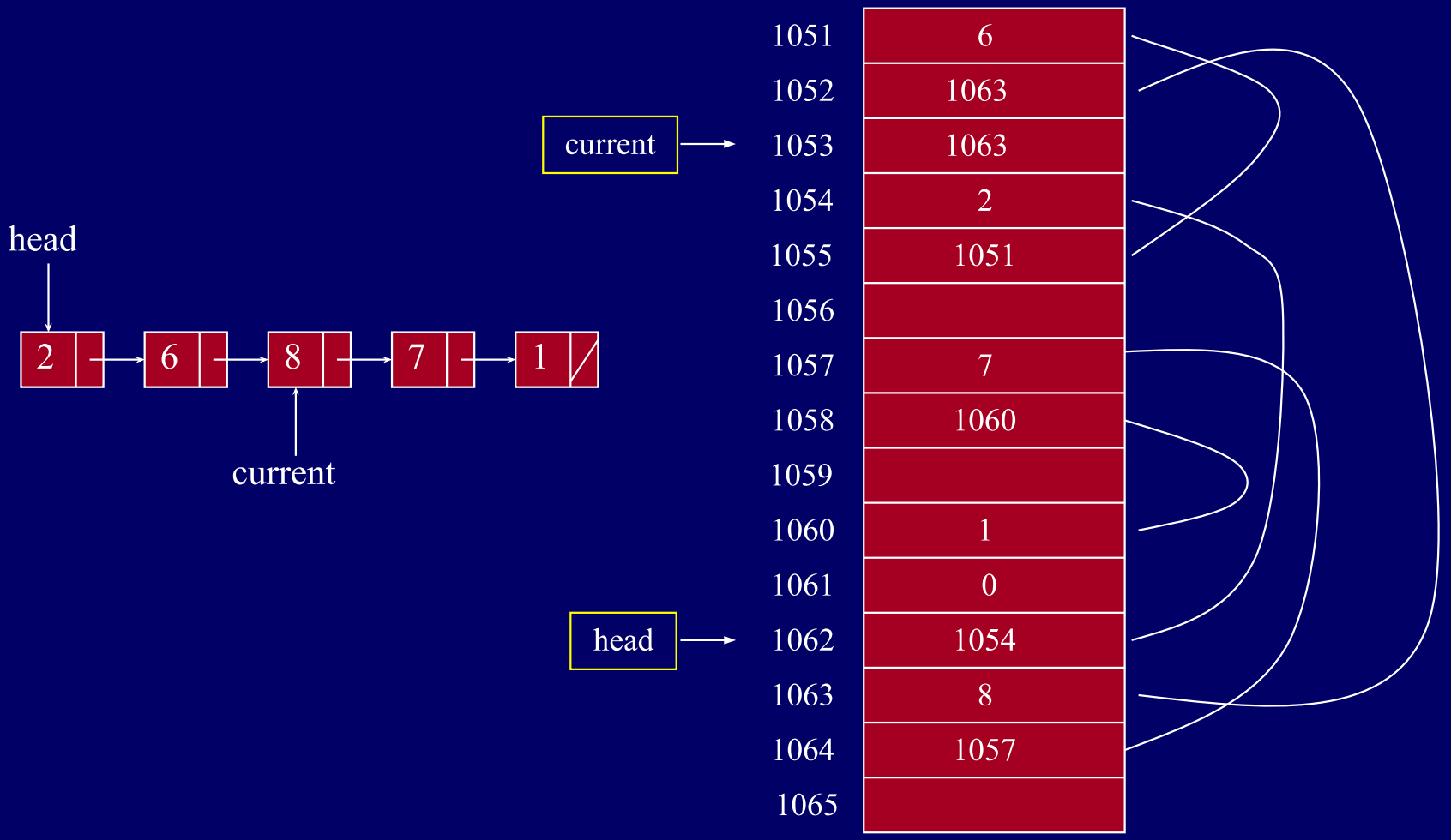
By

Dr Islam Zada

(Lecture 27-28)

# Linked List

- Actual picture in memory:



# Linked List Operations

- add(9): Create a new node in memory to hold ‘9’

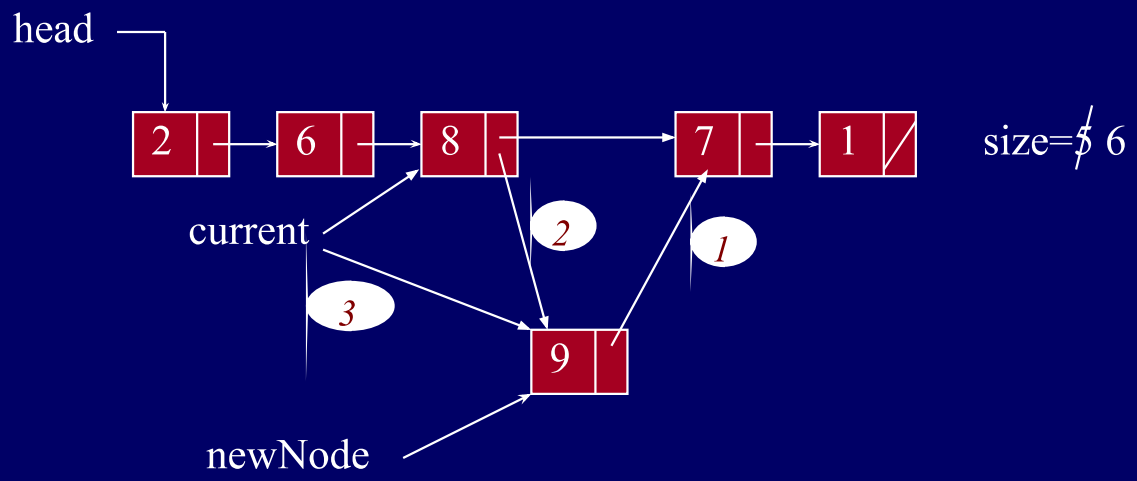
```
Node* newNode = new Node(9);
```

# Linked List Operations

- add(9): Create a new node in memory to hold ‘9’

```
Node* newNode = new Node(9);
```


- Link the new node into the list



# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
           { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
          { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
          { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
           { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
           { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
           { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
           { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
          { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
          { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

*The Node class*

```
class Node {
public:
    int get() { return object; };
    void set(int object) { this->object = object; };

    Node *getNext() { return nextNode; };
    void setNext(Node *nextNode)
          { this->nextNode = nextNode; };
private:
    int object;
    Node *nextNode;
};
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
#include <stdlib.h>
#include "Node.cpp"

class List {
public:
   // Constructor
   List() {
      headNode = new Node();
      headNode->setNext(NULL);
      currentNode = NULL;
      size = 0;
   };
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
          currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# C++ Code for Linked List

```
void add(int addObject) {
     Node* newNode = new Node();
     newNode->set(addObject);
     if( currentNode != NULL ){
         newNode->setNext(currentNode->getNext());
         currentNode->setNext( newNode );
         lastCurrentNode = currentNode;
         currentNode = newNode;
      }
      else {
         newNode->setNext(NULL);
         headNode->setNext(newNode);
         lastCurrentNode = headNode;
         currentNode =  newNode;
      }
      size++;
};
```

# Building a Linked List

```
List list;
```

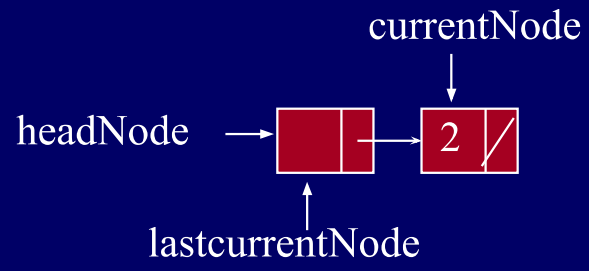

size=0

# Building a Linked List

```
List list;
```


size=0

```
list.add(2);
```



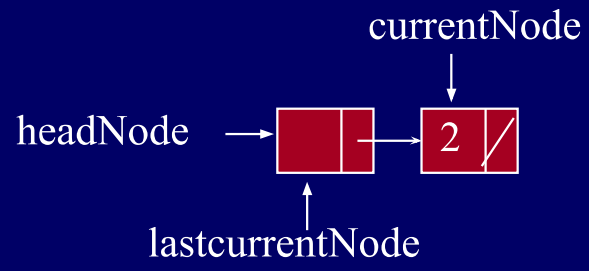

size=1

# Building a Linked List

```
List list;
```

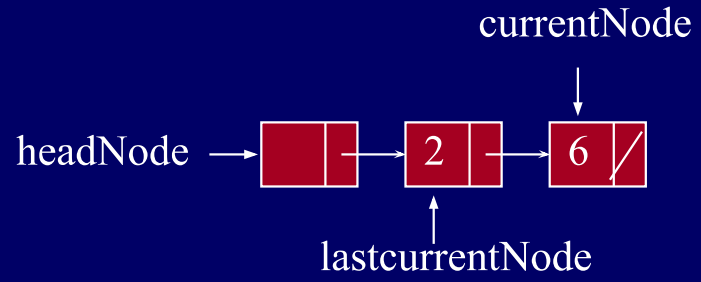

size=0

```
list.add(2);
```




size=1

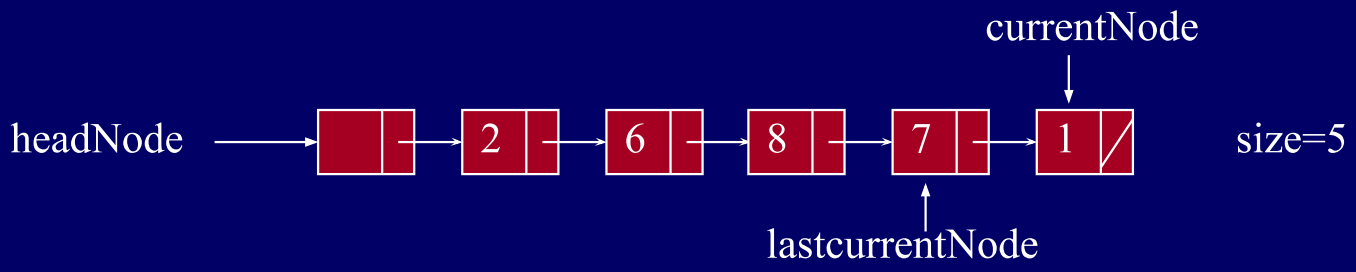
```
list.add(6);
```




size=2

# Building a Linked List

```
List.add(8); list.add(7); list.add(1);
```



# C++ Code for Linked List

```
int get() {
    if (currentNode != NULL)
        return currentNode->get();
};
```

# C++ Code for Linked List

```
bool next() {
    if (currentNode == NULL) return false;

    lastCurrentNode = currentNode;
    currentNode = currentNode->getNext();
    if (currentNode == NULL || size == 0)
      return false;
    else
      return true;
};
```