

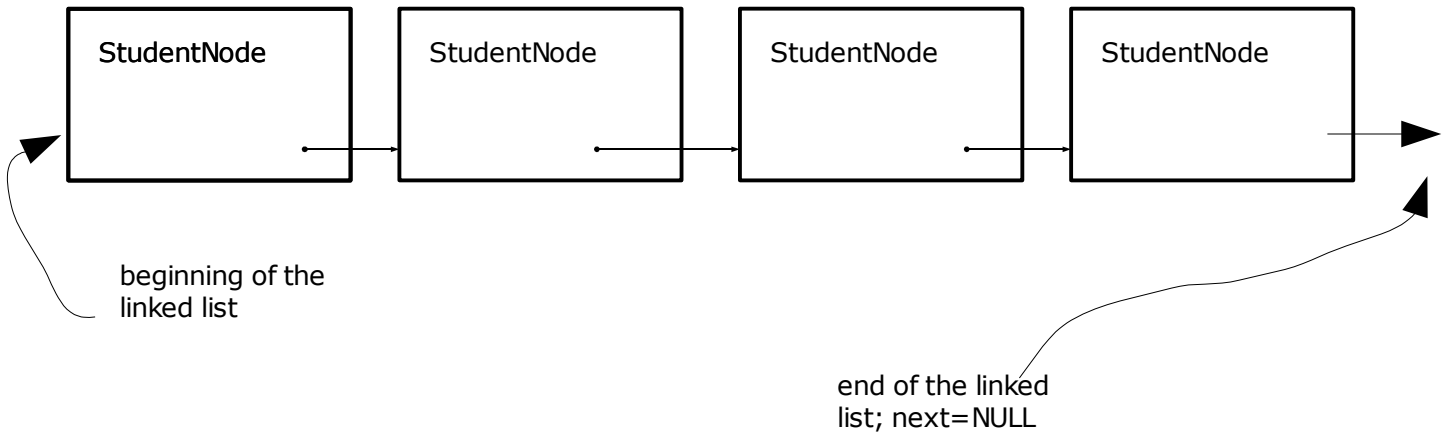
Linked Lists

class Student

```
int ID;  
string name;
```

class StudentNode

```
Student s;  
StudentNode* next;
```



```
//File Student.h
#include<iostream>
#include<string>

using namespace std;
class Student{
public:
    int ID;
    string name;
    Student::Student();
    Student(int, string);
    void display();
};
```

```
//File Student.cpp
#include "Student.h"

Student::Student(int i, string n){
    ID=i;
    name=n;
}
Student::Student(){}
```

```
void Student::display(){
    cout<<"ID = "<<ID<<" Name = "<<name<<endl;
}
```

```
//File StudentNode.h
#include "Student.h"
class StudentNode{
public:
    Student s;
    StudentNode* next;
    StudentNode(Student);
    StudentNode(Student, StudentNode*);
    void display();

};
```

```
//File StudentNode.cpp
#include "StudentNode.h"

StudentNode::StudentNode(Student n){
    s=n;
    next=NULL;
}

StudentNode::StudentNode(Student n, StudentNode* p){
    s=n;
    next=p;
}

void StudentNode::display(){
    s.display();
}
```

```
//File StudentList.h
#include "StudentNode.h"

class StudentList{
public:
    StudentNode* first;
    int size;
    StudentList(StudentNode*);
    StudentList(StudentNode*, int);
    void display();
};
```

```
//File StudentList.cpp
#include "StudentList.h"
StudentList::StudentList(StudentNode* p){
    first=p;
    size=1;
}
StudentList::StudentList(StudentNode* p, int i){
    first=p;
    size=i;
}

void StudentList::display(){
    StudentNode* p;
    p=first;
    while(p != NULL){
        p->display();
        p=p->next;
    }
}

int main(){
    Student s1(1,"Joe P."), s2(2,"Anand S."),s3(3, "Amy W."),s4(5,"Lon
T.");
    StudentNode sn1(s1),sn2(s2), sn3(s3),sn4(s4);
    //Link
    sn1.next=&sn2;
    sn2.next=&sn3;
    sn3.next=&sn4;
    StudentList SL(&sn1,4);
    SL.display();
}
```

```
//Makefile
all: StudentList clean

StudentList: StudentList.cpp StudentNode.o Student.o
    g++ -o StudentList StudentList.cpp StudentNode.o Student.o
StudentNode.o: StudentNode.cpp StudentNode.h Student.o
    g++ -c -o StudentNode.o StudentNode.cpp
Student.o: Student.cpp Student.h
    g++ -c -o Student.o Student.cpp
clean:
    rm -f *.o *.~
```