

Re: can someone please help me with making this class

Re: can someone please help me with making this class

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2006-03/msg01127.html>

- *From:* sparks <jobob@xxxxxxxx>
 - *Date:* Sun, 26 Mar 2006 21:41:50 GMT
-

I guess the weirdest thing is this works.
Not completely because as you pointed out I did
not have the else to get it to completely
insert all the nodes.

Jerry

I made a new project and 2 .h files one node class that I made out of the
struct and another I put the insert in.
And no longer have the errors that I was seeing.

I am not sure if adding the .h to the existing project and doing the same thing
got it confused or what. But I am not finished but it seems to be doing better.

On Sun, 26 Mar 2006 10:49:40 -0800, Barry Schwarz <schwarzb@xxxxxxxx> wrote:

On Sat, 25 Mar 2006 18:37:28 GMT, sparks <jobob@xxxxxxxx> wrote:

I wrote a program for linked list(I have included the insert routine)
this works fine.

Only for some strange definition of fine.

=====

```
struct node
{
string Dname;
string phoneNo;
node *nxt;
};
```

Re: can someone please help me with making this class

```
node *start_ptr = NULL;
node *current;
int option = 0;

//sorted insert in list
void insert_node(string name, string phone)
{

node *temp, *prev ;
temp = new node;
prev = new node;
temp->Drname = name;
temp->phoneNo = phone;
temp->nxt = NULL;
prev->nxt = NULL;
if (start_ptr == NULL)
{
start_ptr = temp;
current = start_ptr;
}
else
{
temp=start_ptr;
```

Didn't you just lose the first structure you created with new and initialized above?

```
prev=NULL;
```

And the second?

```
if (temp->phoneNo < start_ptr->phoneNo)
```

Since temp == start_ptr, can this ever be true?

```
{
temp->nxt = start_ptr;
start_ptr->nxt = NULL;
}
else
{
while((temp->nxt != NULL) && (temp->phoneNo < current->phoneNo))
{
prev->nxt = temp;
```

Re: can someone please help me with making this class

prev was set to NULL. You cannot dereference it. Even if you could, you never update prev in this while loop so you continue to change the same prev->nxt repeatedly.

```
temp = temp -> nxt;
}
temp->nxt = current;
prev->nxt = temp;
}
}
cout<<current->Dname<<endl;
```

Where in the "else block" did current ever get updated?

```
}
```

Remove del for email