```
#include <iostream>
using std::cout;
using std::endl;
using std::ostream;

class list;

class node
{
    int data;
    node *next;

public:
    node()
    {
        data = 0;
        next = NULL;
    }

    ~node()
    {
        data = 0;
        next = NULL;
    }

    friend class list;
    friend ostream& operator <<(ostream &, list &);
};

class list
{
    node *first;

public:
    list()
    {
        first = NULL;
    }

    ~list()
    {
        if(first != NULL)
        {
            delete_all();
            first = NULL;
        }
    }

    void insert_last(int no)
    {
        node *temp = NULL;
        node *newnode = NULL;

        newnode = new node;
        if(NULL == newnode)
        {
            cout << "Memory allocation FAILED\n";
            return;
        }

        newnode->data = no;
        newnode->next = NULL;

        if(NULL == first)
        {
            first = newnode;

```

```
68              return;
69          }
70
71          temp = first;
72          while(temp->next != NULL)
73              temp = temp->next;
74
75          temp->next = newnode;
76      }
77
78      void display()
79      {
80          node *head = first;
81
82          if(NULL == head)
83          {
84              cout << "List is empty\n";
85              return;
86          }
87
88          cout << "list is:\n";
89
90          while(head != NULL)
91          {
92              cout << "|" << head->data << "|->";
93              head = head->next;
94          }
95
96          cout << endl;
97      }
98
99      void delete_all()
100     {
101         node *temp = NULL;
102
103         while(first != NULL)
104         {
105             temp = first;
106             first = temp->next;
107             temp->next = NULL;
108             delete temp;
109         }
110
111         cout << "\nDeleted all nodes successfully\n";
112     }
113
114     friend ostream& operator <<(ostream &, list &);
115 };
116
117 ostream& operator <<(ostream &out, list &lst)
118 {
119         node *head = lst.first;
120
121         if(NULL == head)
122         {
123             out << "List is empty\n";
124             return out;
125         }
126
127         out << "list is:\n";
128
129         while(head != NULL)
130         {
131             out << "|" << head->data << "|->";
132             head = head->next;
133         }
134
```

```
135         cout << endl;
136
137         return out;
138 }
139
140 //---------------------------------
141
142 int main(void)
143 {
144     list obj1;
145     list obj2;
146
147     obj1.insert_last(10);
148     obj1.insert_last(20);
149     obj1.insert_last(30);
150
151     obj2.insert_last(40);
152     obj2.insert_last(50);
153     obj2.insert_last(60);
154
155 /*
156     cout << "First ";
157     obj1.display();
158     cout << "\nSecond ";
159     obj2.display();
160 */
161
162     cout << obj1 << endl << obj2 << endl;
163
164     return 0;
165 }
166
```