

```

1: /*
2:  * C++ Program To Implement Stack using Linked List
3:  */
4: #include<iostream>
5: #include<cstdlib>
6: using namespace std;
7:
8: /*
9:  * Node Declaration
10: */
11: struct node
12: {
13:     int info;
14:     struct node *link;
15: }*top;
16:
17: /*
18:  * Class Declaration
19: */
20: class stack_list
21: {
22:     public:
23:         node *push(node *, int);
24:         node *pop(node *);
25:         void traverse(node *);
26:         stack_list()
27:         {
28:             top = NULL;
29:         }
30: };
31:
32: /*
33:  * Main: Contains Menu
34: */
35: int main()

```

```

36: {
37:     int choice, item;
38:     stack_list sl;
39:     while (1)
40:     {
41:         cout<<"\n-----" <<endl;
42:         cout<<"Operations on Stack" <<endl;
43:         cout<<"\n-----" <<endl;
44:         cout<<"1. Push Element into the Stack" <<endl;
45:         cout<<"2. Pop Element from the Stack" <<endl;
46:         cout<<"3. Traverse the Stack" <<endl;
47:         cout<<"4. Quit" <<endl;
48:         cout<<"Enter your Choice: ";
49:         cin>>choice;
50:         switch(choice)
51:         {
52:             case 1:
53:                 cout<<"Enter value to be pushed into the stack: ";
54:                 cin>>item;
55:                 top = sl.push(top, item);
56:                 break;
57:             case 2:
58:                 top = sl.pop(top);
59:                 break;
60:             case 3:
61:                 sl.traverse(top);
62:                 break;
63:             case 4:
64:                 exit(1);
65:                 break;
66:             default:
67:                 cout<<"Wrong Choice" <<endl;
68:         }
69:     }
70:     return 0;

```

```

71:}
72:
73:/*
74: * Push Element into the Stack
75: */
76:node *stack_list::push(node *top, int item)
77:{
78:    node *tmp;
79:    tmp = new (struct node);
80:    tmp->info = item;
81:    tmp->link = top;
82:    top = tmp;
83:    return top;
84:}
85:
86:/*
87: * Pop Element from the Stack
88: */
89:node *stack_list::pop(node *top)
90:{
91:    node *tmp;
92:    if (top == NULL)
93:        cout<<"Stack is Empty"<<endl;
94:    else
95:    {
96:        tmp = top;
97:        cout<<"Element Popped: " <<tmp->info<<endl;
98:        top = top->link;
99:        delete(tmp);
100:    }
101:    return top;
102:}
103:
104:/*
105: * Traversing the Stack

```

```
106: */
107: void stack_list::traverse(node *top)
108: {
109:     node *ptr;
110:     ptr = top;
111:     if (top == NULL)
112:         cout<<"Stack is empty"<<endl;
113:     else
114:     {
115:         cout<<"Stack elements : "<<endl;
116:         while (ptr != NULL)
117:         {
118:             cout<<ptr->info<<endl;
119:             ptr = ptr->link;
120:         }
121:     }
122: }
```