

```
#include <iostream>
using namespace std;
// Node Structure
struct Node
{
    int data;
    Node *next;
};
// Stack class
class Stack
{
private:
    Node *top;

public:
    // Constructor
    Stack()
    {
        top = NULL;
    }

    // Push Function
    void push(int value)
    {
        Node *newNode = new Node();
        newNode->data = value;
        newNode->next = top;
        top = newNode;
        cout << value << " pushed to stack." << endl;
    }

    // Pop Function
    void pop()
    {
        if (top == NULL)
        {
            cout << "Error: Stack Underflow (Empty)." << endl;
            return;
        }

        Node *temp = top;
        top = top->next;
        cout << temp->data << " popped from stack." << endl;
    }
}
```

```

        delete temp;
    }

// Peek Function
void peek()
{
    if (top == NULL)
    {
        cout << "Stack is Empty." << endl;
        return;
    }
    cout << "Top element is: " << top->data << endl;
}

// Display Function
void display()
{
    if (top == NULL)
    {
        cout << "Stack is Empty." << endl;
        return;
    }

    Node *temp = top;
    cout << "Stack: ";
    while (temp != NULL)
    {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "NULL" << endl;
}
};

// Main Function
int main()
{
    Stack s;
    int choice, val;

    do
    {
        cout << "\n--- DYNAMIC STACK MENU ---" << endl;
        cout << "1. Push" << endl;

```

```
cout << "2. Pop" << endl;
cout << "3. Peek" << endl;
cout << "4. Display" << endl;
cout << "5. Exit" << endl;
cout << "Enter choice: ";
cin >> choice;

switch (choice)
{
case 1:
    cout << "Enter value: ";
    cin >> val;
    s.push(val);
    break;
case 2:
    s.pop();
    break;
case 3:
    s.peek();
    break;
case 4:
    s.display();
    break;
case 5:
    cout << "Exiting..." << endl;
    break;
default:
    cout << "Invalid choice." << endl;
}
} while (choice != 5);

return 0;
}
```