

```
#include <iostream>
using namespace std;
// Size Defination
#define MAX 100
// Stack class
class Stack
{
    int top;
    int arr[MAX];

public:
    // Constructor
    Stack()
    {
        top = -1;
    }

    // Push Function
    void push(int x)
    {
        if (top >= MAX - 1)
        {
            cout << "Error: Stack Overflow! Cannot add more elements." << endl;
            return;
        }
        arr[++top] = x;
        cout << x << " pushed into stack." << endl;
    }

    // Pop Function
    void pop()
    {
        if (top < 0)
        {
            cout << "Error: Stack Underflow! Stack is empty." << endl;
            return;
        }
        int x = arr[top--];
        cout << x << " popped from stack." << endl;
    }

    // Peek Function
    void peek()
```

```

{
    if (top < 0)
    {
        cout << "Stack is Empty." << endl;
        return;
    }
    cout << "Top element is: " << arr[top] << endl;
}

// Display Function
void display()
{
    if (top < 0)
    {
        cout << "Stack is Empty." << endl;
        return;
    }
    cout << "Stack elements: ";
    for (int i = top; i >= 0; i--)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}
};

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

    do
    {
        cout << "\n--- STACK OPERATIONS ---" << endl;
        cout << "1. Push" << endl;
        cout << "2. Pop" << endl;
        cout << "3. Peek (Top)" << endl;
        cout << "4. Display" << endl;
        cout << "5. Exit" << endl;
        cout << "Enter choice: ";
        cin >> choice;

        switch (choice)

```

```
{  
case 1:  
    cout << "Enter value to push: ";  
    cin >> value;  
    s.push(value);  
    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;  
}
```