Assignment 3 Question 1

#include <iostream>

using namespace std;

#define SIZE 5 // maximum size of stack

class Stack {

int arr[SIZE];

int top;

public:

Stack() { top = -1; }

// check if stack is empty

bool isEmpty() {

return (top == -1);

}

// check if stack is full

bool isFull() {

return (top == SIZE - 1);

}

// push element into stack

void push(int x) {

if (isFull()) {

cout << "Stack Overflow! Cannot push " << x << endl;

} else {

arr[++top] = x;

cout << x << " pushed into stack.\n";

}

}

// pop element from stack

void pop() {

if (isEmpty()) {

cout << "Stack Underflow! Nothing to pop.\n";

} else {

cout << arr[top--] << " popped from stack.\n";

}

}

// peek top element

void peek() {

if (isEmpty()) {

cout << "Stack is Empty!\n";

} else {

cout << "Top element: " << arr[top] << endl;

}

}

// display stack elements

void display() {

if (isEmpty()) {

cout << "Stack is Empty!\n";

} else {

cout << "Stack elements: ";

for (int i = top; i >= 0; i--) {

cout << arr[i] << " ";

}

cout << endl;

}

}

};

// main menu

int main() {

Stack s;

int choice, value;

do {

cout << "\n--- Stack Menu ---\n";

cout << "1. Push\n2. Pop\n3. Peek\n4. isEmpty\n5. isFull\n6. Display\n7. Exit\n";

cout << "Enter your 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:

if (s.isEmpty())

cout << "Stack is Empty.\n";

else

cout << "Stack is NOT Empty.\n";

break;

case 5:

if (s.isFull())

cout << "Stack is Full.\n";

else

cout << "Stack is NOT Full.\n";

break;

case 6:

s.display();

break;

case 7:

cout << "Exiting program...\n";

break;

default:

cout << "Invalid choice! Try again.\n";

}

} while (choice != 7);

return 0;

}