جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني

Department of Cyber Security

Subject: Data Structure

Class: Second

Lecturer: Asst. Prof. Dr. Ali Kadhum Al-Quraby

Lecture: (10)

Implement a stack

Study Year: 2024-2025

Basic Stack Operations (Without STL Stack)

```
#include <iostream>
#define MAX 100
using namespace std;

struct Stack {
    int arr[MAX];
    int top;
};

// Initialize the stack
void initStack(Stack &s) {
    s.top = -1;
}

// Check if the stack is empty
bool isEmpty(Stack &s) {
    return s.top == -1;
}

// Push an element onto the stack
void push(Stack &s, int value) {
    if (s.top == MAX - 1) {
        cout << "Stack overflow!" << endl;
```

```
        return;
    }
    s.arr[++s.top] = value;
}

// Pop an element from the stack
int pop(Stack &s) {
    if (isEmpty(s)) {
        cout << "Stack underflow!" << endl;
        return -1;
    }
    return s.arr[s.top--];
}

// Get the top element of the stack
int top(Stack &s) {
    if (isEmpty(s)) {
        cout << "Stack is empty!" << endl;
        return -1;
    }
    return s.arr[s.top];
}

// Get the size of the stack
int size(Stack &s) {
    return s.top + 1;
```

```
}

int main() {
    Stack s;
    initStack(s);

    // Push elements onto the stack
    for (int i = 1; i <= 5; i++) {
        push(s, i);
    }

    // Display the top element
    cout << "Top element: " << top(s) <<
endl;

    // Pop all elements and display them
    cout << "Stack elements: ";
    while (!isEmpty(s)) {
        cout << pop(s) << " ";
    }
    cout << endl;

    return 0;
}
```

Reverse a String

```
cpp
Copy code
#include <iostream>
#define MAX 100
using namespace std;

struct Stack {
    char arr[MAX];
    int top;
};

// Initialize the stack
void initStack(Stack &s) {
    s.top = -1;
}

// Push a character onto the stack
void push(Stack &s, char value) {
    if (s.top == MAX - 1) {
        cout << "Stack overflow!" << endl;
        return;
    }
    s.arr[++s.top] = value;
}
```

```
// Pop a character from the stack
char pop(Stack &s) {
    if (s.top == -1) {
        cout << "Stack underflow!" << endl;
        return '\0';
    }
    return s.arr[s.top--];
}

int main() {
    string str = "hello";
    Stack s;
    initStack(s);

    // Push all characters onto the stack
    for (char c : str) {
        push(s, c);
    }

    // Pop characters to reverse the string
    string reversed = "";
    while (s.top != -1) {
        reversed += pop(s);
    }
```

```
    cout << "Reversed string: " << reversed
<< endl;

    return 0;
}
```

Check for Balanced Parentheses

```
cpp
Copy code
#include <iostream>
#define MAX 100
using namespace std;

struct Stack {
    char arr[MAX];
    int top;
};

// Initialize the stack
void initStack(Stack &s) {
    s.top = -1;
}

// Push a character onto the stack
```

```
void push(Stack &s, char value) {
    if (s.top == MAX - 1) {
        cout << "Stack overflow!" << endl;
        return;
    }
    s.arr[++s.top] = value;
}

// Pop a character from the stack
char pop(Stack &s) {
    if (s.top == -1) {
        cout << "Stack underflow!" << endl;
        return '\0';
    }
    return s.arr[s.top--];
}

// Check if the string is balanced
bool isBalanced(string str) {
    Stack s;
    initStack(s);

    for (char c : str) {
        if (c == '(' || c == '{' || c ==
'[') {
       push(s, c);
```

```
        }
else if (c == ')' || c == '}' || c == ']')
{
    if (s.top == -1) return false;
        char topChar = pop(s);
        if ((c == ')' && topChar != '(') ||
            (c == '}' && topChar != '{') ||
            (c == ']' && topChar != '[')) {
              return false;
            }
         }
    }
    return s.top == -1;
}

int main() {
    string str = "{[()()]}";
    if (isBalanced(str)) {
        cout << "Balanced" << endl;
    } else {
        cout << "Not Balanced" << endl;
    }
    return 0;
}
```

Convert Decimal to Binary

```
cpp
Copy code
#include <iostream>
#define MAX 100
using namespace std;

struct Stack {
    int arr[MAX];
    int top;
};

// Initialize the stack
void initStack(Stack &s) {
    s.top = -1;
}

// Push an integer onto the stack
void push(Stack &s, int value) {
    if (s.top == MAX - 1) {
        cout << "Stack overflow!" << endl;
        return;
    }
    s.arr[++s.top] = value;
}
```

```
// Pop an integer from the stack
int pop(Stack &s) {
    if (s.top == -1) {
        cout << "Stack underflow!" << endl;
        return -1;
    }
    return s.arr[s.top--];
}

int main() {
    int num = 10;
    Stack s;
    initStack(s);

    // Convert decimal to binary
    while (num > 0) {
        push(s, num % 2);
        num /= 2;
    }

    // Print binary representation
    cout << "Binary: ";
    while (s.top != -1) {
        cout << pop(s);
    }
```

```
    cout << endl;

    return 0;
}
```