***Assignment 2 Data Structures***

***Adeeba Nizam***

**README.md / Report**

**Q)2 Stack of Integers using Vector**

This program implements a **stack of integers** using the C++ std::vector container.

It includes all major stack operations:

* Pushing elements onto the stack (push)
* Removing elements from the stack (pop)
* Checking if the stack is empty (empty)
* Getting the top element (top)
* Calculating the average of all elements (average)

### **Files Included:**

|  |  |
| --- | --- |
| * Stack.h | Header file containing class definition and function declarations |
| * Stack.cpp | Implementation of all stack operations |
| * main.cpp | Driver file to test all stack functionalities |
| * README.md | Report and running instructions |

### **File 1: Stack.h**

This is the **header file** that defines the structure of the Stack class.  
It includes:

* Private member: vector<int> data (to store elements)
* Public member functions for stack operations.

|  |  |
| --- | --- |
| void push(int value) | Adds an integer to the top of the stack. |
| void pop() | Removes the top element if the stack is not empty. |
| int top() const | Returns the top element of the stack. |
| bool empty() const | Checks if the stack is empty. |
| double average() const | Computes and returns the average of the elements. |
| void display() const | Prints all elements from bottom to top. |
| **File 2: Stack.cpp** This file provides **implementation** for each declared function. Each function is well-commented to explain its purpose and behavior. |  |

### **File 3: main.cpp**

This is the **driver file** that tests all the functions.

1. Creates a stack object (Stack s;)
2. Checks if the stack is empty
3. Pushes several integers (10, 20, 30, 40, 50)
4. Displays all stack elements
5. Pops the top element
6. Shows the current top element
7. Calculates and prints the average of stack elements
8. Displays the final state of the stack

Running the code In Visual Studios:

Make sure all the files are in the same directory:

* Stack.h
* Stack.cpp
* main.cpp

