Data Structures and Algorithms Lab

Instructions

Work on this lab individually. *Write **main** function first and keep on testing the functionality of each function once created.*

Program the following tasks in your **C++** compiler and then compile and execute them.

Email your solution (**.cpp**) file only to the following respective recipient till **Friday, April 02, 2021**.

DO NOT compress/zip your solution.

The email must be sent from your **official PUCIT email id**, otherwise it will **NOT BE ACCEPTED** and will be marked **ZERO**.

The subject of the email should be the exact name of the lab i.e. **Lab 05**. **2 MARKS** will be **DEDUCTED**, otherwise.

Degree	Recipient Email	Subject of Email
BSIT Morning	dsaubt01@gmail.com	Lab 05
BSIT Afternoon	dsaubt02@gmail.com	

You are strictly not allowed to add any other data-member/constructor/function in the class. You are also not allowed to change the name or prototype of any data-member/constructor/function.

ADT: Stack

Provide the implementation of the following generic **Stack** class; it should provide the standard stack structure of LIFO (last in first out) as discussed in the class.

```
template <class T>
class Stack
{
public:
	//constructor to create MAX_SIZE stack dynamically
	Stack(int MAX_SIZE);

	//destructor to free any memory resources occupied by stack
	~Stack();

	//stack manipulation operations
	void push(T newItem);		//push new item
	T pop();			//pop item
	T Top();			//return the item at the top
	void clear ();			//clear the stack

	//stack status operations
	bool isEmpty();			//is stack empty?
	bool isFull();			//is stack full?

	//outputs the data in stack. If the list is empty, outputs "Empty Stack".
	void showStructure() const;

private:
	//Data members
	T *data;			//array of data items allocated dynamically based on MAX_SIZE
	int top;			//top of the stack
	const int MAX_SIZE;		//maximum capacity of the stack
};
```

The show structure function must display the stack status with its top pointing to the correct location on the console.

Sample run:

```
stack.Push(5.0);
stack.Push(6.5);
stack.showStructure();
```

```
top --> 6.5
        5
```

```
stack.Push(-3.0);
stack.Push(-8.0);
stack.showStructure();
```

```
top --> -8
        -3
        6.5
        5
```

```
stack.Pop();
stack.Pop();
stack.showStructure();
```

```
top --> 6.5
        5
```

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺