1. Implement Following Operations

1)

Stack (int ignored = 0)

Requirements: None

Results: Constructor. Creates an empty stack.

2)

~Stack ()

Requirements: None

Results: Destructor. Deallocates (frees) the memory used to store a stack.

3)

void push (const DataItem)

Requirements: None

Results: Push the element at top of the stack.

4)

Void pop ()

Requirements: Stack is not empty Results: Returns the

element from the top of the stack.

5)

element Peek ()

return element at the top of stack

6)

void clear ()

Requirements: None

Results: Removes all the elements from a stack.

7)

Bool isEmpty ()

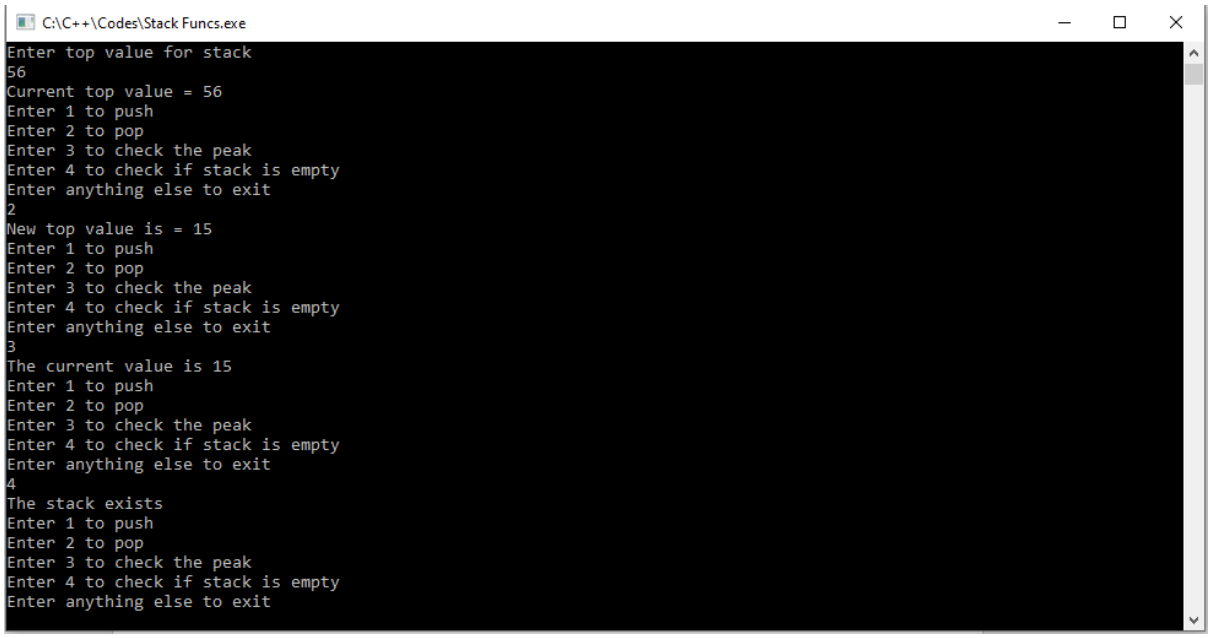
Requirements: None

Results: Returns true if a stack is empty. Otherwise, returns false

```
#include <iostream>
using namespace std;
class Stack{
	private:
	int stack[1000];
	int top=-1;
	public:
		void push(){
			top++;
			if(top<1001){



			cout<<"Enter top value for stack"<<endl;
		cin>>stack[top];
		cout<<"Current top value = "<<stack[top]<<endl;
		}
		else{
			cout<<"Stack is overloaded"<<endl;
		}}
		void pop(){
			top--;
			if(top<-1){
				top=-1;
			}
			if(top>=0){



		cout<<"New top value is = "<<stack[top]<<endl;
		}
		else{
```

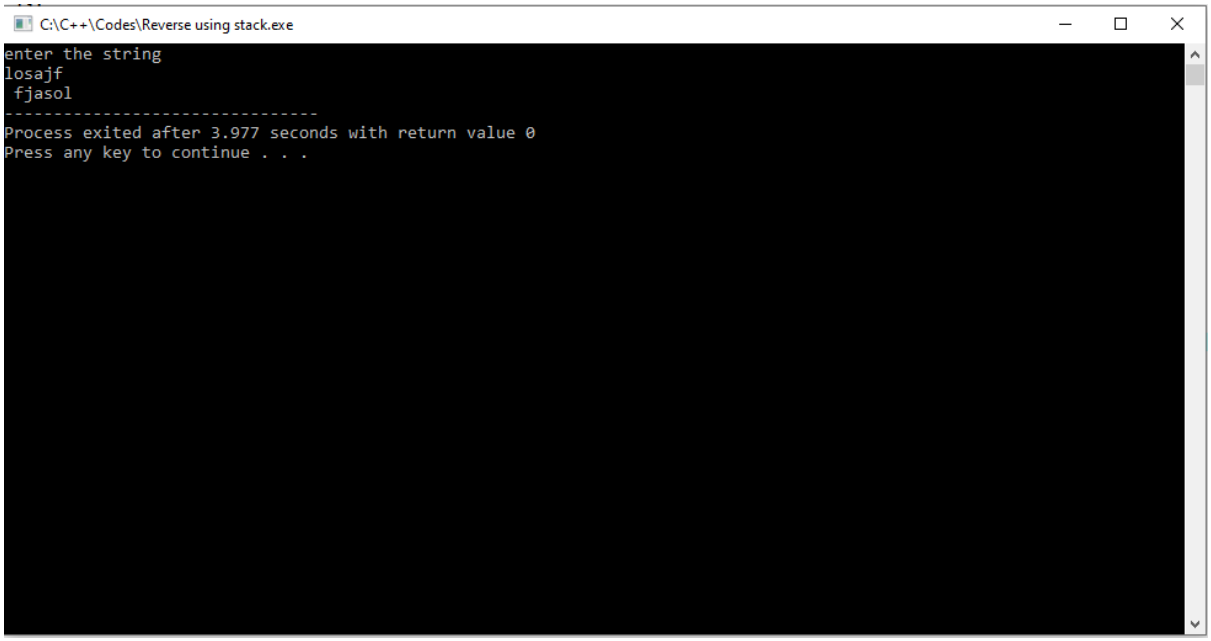
```
		cout<<"Stack does not exist"<<endl;}
		}
		void peak(){
			if(top>=0){

			cout<<"The current value is "<<stack[top]<<endl;
		}
		else{
			cout<<"The stack does not exist"<<endl;
		}}
		void empty(){
			if(top<0){
				cout<<"The stack does not exist"<<endl;
			}
			else{
				cout<<"The stack exists"<<endl;
			}
		}
};

int main(){
	Stack st;
	label:
	int ch=0,loop=0;

	cout<<"Enter 1 to push"<<"\nEnter 2 to pop"<<"\nEnter 3 to check the peak"<<"\nEnter 4 to check if stack is empty\n"<<"Enter anything else to exit\n";
	cin>>ch;
	switch(ch){
	case 1:
	st.push();
```

```
        break;
        case 2:
        st.pop();
        break;
        case 3:

        st.peak();
        break;
        case 4:
        st.empty();
        break;
        default:
                loop=1;
                break;
}
if(loop==0){
        goto label;
}
}
```



2. Write a program in C++ to reverse a string (Data Structures) using stack.

```
#include <iostream>
#include <string>
using namespace std;
class STACK{
	private:
		string ma;
	int top=-1,max;
	public:
	void inp(){
		cout<<"enter the string"<<endl;
		cin>>ma;
	top=ma.size();
	}
	void show(){
		for(top;top>-1;top--){
			cout<<ma[top];
		}
	}

};
int main(){
	STACK sa;
sa.inp();
sa.show();

}
```


C:\C++\Codes\Reverse using stack.exe
enter the string
losajf
 fjasol

Process exited after 3.977 seconds with return value 0
Press any key to continue . . .