Template functions and classes are generic classes

They can work for any type of data
Classes can support different type of data

```
template<class T>
class Stack
{
private:
    T *stk;
    int top;
    int size;
public:
    Stack(int sz)
    {
        size=sz;
        top=-1;
        stk=new T[size];
    }
    void push(T x);
    T pop();
};

template<class T>
void Stack<T>::push(T x)
{
    if(top==size-1)
        cout<<"Stack is Full";
    else
    {
        top++;
        stk[top]=x;
    }
}

template<class T>
T Stack<T>::pop()
{
    T x=0;
    if(top==-1)
        cout<<"Stack is Empty"<<endl;
    else
    {
        x=stk[top];
        top--;
    }
    return x;
}

int main()
{
    Stack<float> s(10);
    s.push(10);
    s.push(23);
    s.push(33);
}
```