

CODE SAMPLE: GENERIC STACK

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;

namespace GenericStack
{
    class Program
    {
        static void Main(string[] args)
        {
            MyStack<int> s = new MyStack<int>();
            s.Push(19);
            s.Push(28);
            Console.WriteLine("s.Contains(17) is {0}", s.Contains(17));
            Console.WriteLine("s.Contains(19) is {0}", s.Contains(19));
            Console.WriteLine("s.Contains(0) is {0}", s.Contains(0));
            int x = s.Pop();
            int y = s.Pop();
            Console.WriteLine("x is {0} and y is {1}", x, y);
            // MyStack<Hashtable> s2 = new MyStack<Hashtable>(); //Won't
            // compile because Hashtables aren't comparable
        }
    }
    class MyStack<T> where T:IComparable<T>
    {
        private T[] _itemsArray;
        private int _index;
        public const int MAX_SIZE = 100;
        public MyStack()
        {
            _itemsArray = new T[MAX_SIZE];
            _index = 0;
        }

        public void Push(T item)
        {
            if (_index == MAX_SIZE)
                throw new System.StackOverflowException("Stack is full.");
            _itemsArray[_index++] = item;
        }

        public T Pop()
        {
            if (_index == 0)
                throw new System.InvalidOperationException("Stack is empty.");
            return _itemsArray[--_index];
        }

        public bool Contains(T t)
        {
            for( int i = 0; i < _index; i++)
            {
                if (_itemsArray[i].CompareTo(t) == 0) return true;
            }
            return false;
        }
    }
}
```