

CS 2026 – Spring 2009
Assignment #2
2/1/2009
Due: Friday 2/6/2009 11:59 PM

Update: As a simple optimization step, whenever a `Contains(T item)` call is made, the item referenced should be moved to the front of the list used to store the elements.

This assignment will exercise your knowledge of C# features such as enumerators, generics, operator overload, properties, and delegates that were discussed last week.

In this assignment you will implement a generic `Set<T>` class that is parameterized by the type of object it contains. The set will be used as a basic container of objects of type T. Since this is a set, there is only one copy of any object in it. For example: `{1,2,2,3}` and `{1,2,3}` are the same set.

Methods

Your `Set<T>` should implement the following methods:

- `Set()`
 - The constructor for an empty set.
- `Set(IEnumerable<T> e)`
 - A constructor that fills the set with all the elements in some enumerable collection.
- `public static Set<T> operator +(Set<T> lhs, Set<T> rhs)`
 - This operator implements set union: it should return a new set that contains any item contained in either the `lhs` or the `rhs` set.
- `public static Set<T> operator -(Set<T> lhs, Set<T> rhs)`
 - This operator implements set difference: it should return a new set containing all the elements that were in the `lhs` set but not in the `rhs` set.
- `public static Set<T> operator *(Set<T> lhs, Set<T> rhs)`
 - This operator implements set intersection: it should return a new set containing all the elements that were in both sets.
- `public static bool operator ==(Set<T> lhs, Set<T> rhs)`
 - This operator returns true if the two sets have exactly the same elements in them.
- `public bool Add(T item)`
 - This method adds the input element to the set. It returns true if the element is added to the set, and false if the element is already present in the set.
- `public bool Remove(T item)`
 - This method removes the input element from the set. It returns true if the element is removed to the set, and false if the element was not in the set to begin with.
- `public bool Contains(T item)`
 - This method returns true if and only if the input element is in the set.
- `public Set<T> Filter(F<T> filterFunction)`
 - This method takes a delegate of type `bool F<T>(T elt)` and returns all the elements in the set for which this function returns true.
- Sets should be enumerable by `foreach`.

Properties

Your `Set<T>` should implement the following properties:

- `public int Count`
 - This property returns the number of elements in the set. This property should be readable but not writable.
- `public bool IsEmpty`
 - This property returns `true` if there are no items in the set. This property should be readable but not writable.
- `public bool Freeze`
 - This property should default to `false`. If this property is set to `true`, then the current set object should ignore calls to `Add(T)` an element to it. (You do not have to raise an exception, you can accept the call, but just do not add that element to your set). This property should be readable and writable.

Other Directions

- To avoid possible name collisions, define your `Set<T>` class in namespace `AssignmentTwo`.
- You are free to add any helper functions, data types ..etc to your code
- You should not base your implementation any of C#'s existing `Set/HashSet` constructs or any other `Set` construct in C#.

Tip

I recommend using C#'s `List<T>` class to store the data in your set. You can find more information about it on MSDN: <http://msdn.microsoft.com/en-us/library/6sh2ey19.aspx>

Why is this useful?

In addition to exercising your skills in a variety of C# features, this assignment will give you first hand experience in creating data structures in C#. Although C# has many data structures to contain data, it is sometimes necessary to create your own custom data structure to provide you with some special functionality that is present in the default ones.

Academic Integrity Reminder

Remember that you may have general discussions about how to approach this problem with your peers, but you should work on the final solution by yourself alone. If you are stuck or are having trouble, you may email me or talk to me after class on Monday or during my office hour on Wednesday.

Good Luck!