

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

namespace SimpleDelegateExample
{
    class Subject
    {
        // Create delegate type "notifier"
        public delegate void notifier(string message);

        // Create instance of "notifier" type, called myNotifier
        public notifier myNotifier;

        public string subjectContents;

        public string SubjectContents
        {
            get
            {
                return subjectContents;
            }

            // Set accessor. First sets the value of the Subject's string (subjectContents)
            // then notifies listeners by calling the delegate myNotifier. All listeners
            // that have registered with Subject (by adding a method to myNotifier) are
            // called.
            set
            {
                subjectContents = value;

                // Check to see that there is at least one method in the delegate instance
                // then call the delegate (i.e., call all methods referenced in the
                delegate)
                if (myNotifier != null)
                {
                    myNotifier(value);
                }
            }
        }
    }

    class Observer
    {
        public string observername;

        // Constructor
        // Saves the name of the observer (we use this in printed output when we receive a
        notification)
        // Adds receive_notification method to myNotifier delegate instance found in
        subject
        public Observer(string name, Subject subject)
        {
            observername = name;
            subject.myNotifier += receive_notification;
        }

        public void receive_notification(string message)
        {
            System.Console.WriteLine("{0}: received notification: {1}", observername,
            message);
        }

        public void subscribe(Subject subject)
        {

```

```
        subject.myNotifier += receive_notification;
    }

    public void unsubscribe(Subject subject)
    {
        subject.myNotifier -= receive_notification;
    }
}
//

class Program
{
    static void Main(string[] args)
    {
        Subject mySubject = new Subject();
        Observer bob = new Observer("Bob", mySubject);
        Observer alice = new Observer("Alice", mySubject);
        // Just create reference for now, will create instance later
        Observer maria;
        Subject secondSubject;

        mySubject.SubjectContents = "Hello, eventful world!";
        System.Console.WriteLine();
        mySubject.SubjectContents = "Oooh, baby!";
        System.Console.WriteLine();

        // Create new Observer, maria, and unsubscribe bob.
        maria = new Observer("Maria", mySubject);
        bob.unsubscribe(mySubject);

        mySubject.SubjectContents = "Who can it be now?";
        System.Console.WriteLine();

        // Create second Subject, and have Observers subscribe to it
        secondSubject = new Subject();

        bob.subscribe(secondSubject);
        alice.subscribe(secondSubject);
        maria.subscribe(secondSubject);

        secondSubject.SubjectContents = "Hello, I am the second Subject";
        System.Console.WriteLine();
        mySubject.SubjectContents = "First subject here";

        System.Console.WriteLine("That's all folks");
    }
}
}
```