Ryne Bell CM 438
CSSE 376
Lab 03

1. The project source code is found in the Expedia Project.

2. The classes in Expedia are Car, Flight, Hotel, User, and Booking.

3. The flight class contains functionality to describe information about a flight (it's startDate, endDate, the number of miles), and a method to compare it to other objects (including flight objects) to determine if they are 'Equal'.

4. The test classes in the project are BookingTest, CarTest, FlightTest, HotelTest, and UserTest.

5. The test methods in the UserTest class are TestThatUserInitializes(), TestThatUserHasZeroFrequentFlierMilesOnInit(), TestThatUserCanBookEverything(), TestThatUserHasFrequentFlierMilesAfterBooking(), TestThatUserCanBookAFlight(), TestThatUserCanBookAHotelAndACar(), and TestThatUserHasCorrectNumberOfFrequentFlyerMilesAfterOneFlight().

6. These are some of the functions that are supported by the Assert class: AreEqual(object, object), AreNotEqual(object, object), AreSame(object, object), AreNotSame(object, object, string), Fail().

7. AreEqual: Verifies that two objects are equal. Two objects are considered equal if both are null, or if both have the same value.
AreNotEqual: Verifies that two objects are not equal. Two objects are considered equal if both are null, or if both have the same value.
AreSame: Asserts that two objects refer to the same object.
AreNotSame: Asserts that two objects do not refer to the same object.
Fail: Throws an NUnit.Framework.AssertionException. (Automatically fails the assertion)

8. AreEqual checks the value of the objects to determnine of they are 'equal', AreSame checks to see if both objects are in fact the same object.

9. The TestThatHotelInitalizes unit test first initalizes an object of type Hotel, then ensures that it is not null, and has therefor been initialized.

10. The generic algorithm for calulating the base price is 45 * numberOfNightsToRent.

11. These new tests check if the correct basePrice is calculated for 1, 2, and 10 night stays.

12. We don't need an IsNotNull check because we have already tested that in a previously created unit test, and need not check it again, in fact that would only make understaning what exactly each unit test checks more difficult to understand.

13. The constructor for Hotel is supposed to throw an exception of type ArgumentOutOfRangeException in the event that a value <=0 is passed into the constructor.

14. After the [Test()] included in the tests run by NUnit, I would place [ExpectedException(typeof(OutOfMemoryException)] to specifiy the expected exception.