Alex Jacoby

1. The source code is found in the Expedia project and the test code is in the ExpediaTest project.
2. The classes in the project are: AssemblyInfo, Booking, Car, Flight, Hotel, and User.
3. The flight class allows for tracking of airplane flights, their arrival and departure dates along with the miles traveled.
4. The test classes are BookingTest, CarTest, FlightTest, HotelTest and UserTest.
5. The test methods in UserTest are TestThatUserInitializes, TestThatUserHasZeroFrequentFlierMilesOnInit, TestThatUserCanBookEverything, TestThatUserHasFrequentFlierMilesAfterBooking, TestThatUserCanBookAFlight, TestThatUserCanBookAHotelAndACar, and TestThatUserHasCorrectNumberOfFrequentFlyerMilesAfterOneFlight.
6. Assert.AreEqual, Assert.Less, Assert.False all use Assert.
7. Assert.AreEqual tests to see if two objects are equal (are both null or have the same value). Assert.Less tests verifies if the first value is less than the second value. Assert.False asserts that a condition is false.
8. AreEqual checks if the two objects are equal, AreSame checks to see if the two objects being compared are the exact same (i.e. references to the same object).
9. The unit test is verifying that when creating a hotel object, something is actually created.
10. The generic algorithm is 45*nightsToRent.
11. The cases that are tested are nightsToRent = 1, nightsToRent =2, and nightsToRent = 10.
12. Because if the target is null, the tests will fail anyway.
13. This test expects an exception because we are passing a negative number of nights to stay at a hotel, which is impossible. In hotel.cs, there is a check to see if the nights to stay at a hotel is greater than 0, and if the number of nights isn't greater than 0 it throws an exception.
14. [Test()][ExpectedException(typeof(OutOfMemoryException))]