Lab 3

Daniel Schepers

1) The source code is found in Expedia. The test code is found in ExpediaTest.
2) Expedia contains Car, Flight, Hotel, and User.
3) A Flight contains an arrival and departure time, as well as a distance travelled. The flight class allows us to see whether two flights are equal by comparing the aforementioned values.
4) There are five test classes: BookingTest, CarTest, FlightTest, HotelTest, and UserTest.
5) UserTest contains TestThatUserInitializes, TestThatUserHasZeroFrequentFlierMilesOnInit, TestThatUserCanBookEverything, TestThatUserHasFrequentFlierMilesAfterBooking, TestThatUserCanB ookAFlight, TestThatUserCanBookAHotelAndCar, and TestThatUserHasCorrectNumberOfFrequentFlyerMilesAfterOneFlight.
6) AreEqual(Object Expected, Object Actual), AreNotEqual(Object Expected, Object Actual), AreSame(Object Expected, Object Actual)
7) AreEqual(Object Expected, Object Actual) tests to see whether two objects have equivalent fields. AreNotEqual(Object Expected, Object Actual) test to see that two objects do not have equivalent fields. AreSame(Object Expected, Object Actual) tests to see whether both objects are the same (referencing the same object).
8) AreEqual tests to see whether two objects have the same values: AreSame tests to see whether two objects are the same object.
9) The test verifies that the Hotel class successfully creates a hotel with the given number of nights to rent.
10) getBasePrice works by taking 45*numberOfNightsToRent.
11) The prices of stays of lengths 1, 2, and 10 are calculated.
12) We have already tested that the Hotel constructor works correctly in a previous test.
13) In Hotel.cs, there is an if statement that throws an exception if nightsToRent<=0.
14) [ExpectedException(typeof(OutOfMemoryException))]