Q1: The source code is found in Expedia, and the test code in is ExpediaTest

Q2: AssemblyInfo, Booking, Car, Flight, Hotel, User

Q3: The creation of a flight trasportation, including its details such as start and ending datetime and the amount of miles it covers, and a way to check if such flights are equal

Q4: BookingTest, CarTest, FlightTest, HotelTest, and UserTest

Q5: TestThatUserInitializes, TestThatUserHasZeroFrequentFlierMilesOnInit, TestThatUserCanBookEverything, TestThatUserHasFrequentFlierMilesAfterBooking

Q6: AreEqual, False, IsEmpty

Q7: AreEqual tests if objects are equal (meaning either both null or have same values), False tests if the condition is false, IsEmpty tests that the collection is empty

Q8: AreEqual tests if the values in the object are the same or equal, while AreSame tests to see if both objects are actually the same object

Q9: It is verifying that the Hotel is initialized properly from the input of the days it will be rented out.

Q10: Days Rented * 45

Q11: Tests to make sure that varying inputs for Days Rented in Hotel actually create the desired BasePrice.

Q12: We already tested that passing in numbers as parameters works properly in a previous test, it is no longer helpful to continue testing that functionality

Q13: A hotel cannot be rented for a negative amount of days

Q14: [ExpectedException(typeof(OutOfMemoryException))]