

**[Question 1 (1 pt)]**

In what project is the source code found? In what project is the test code found?

- (i) Expedia
- (ii) ExpediaTest

**[Question 2 (2 pts)]**

What classes exist in the project?

[Does not consider the Test Classes since they're part of question 4]

AssemblyInfo.cs  
Booking.cs  
Car.cs  
Flight.cs  
Hotel.cs  
User.cs

**[Question 3 (4 pts)]**

Explain, in your own words, the functionality that is supported by the Flight class.

Sets a user's Flight info based upon flight dates and distance—for use with Booking info.

**[Question 4 (2 pts)]** What are the test classes in the project?

BookingTest.cs  
CarTest.cs  
FlightTest.cs  
HotelTest.cs  
UserTest.cs

**[Question 5 (4 pts)]** Identify the various test methods in the UserTest class.

TestThatUserInitializes()  
TestThatUserHasZeroFrequentFlierMilesOnInit()  
TestThatUserCanBookEverything()  
TestThatUserHasFrequentFlierMilesAfterBooking()  
TestThatUserCanBookAFlight()  
TestThatUserCanBookAHotelAndACar()  
TestThatUserHasCorrectNumberOfFrequentFlyerMilesAfterOneFlight()

**[Question 6 (4 pts)]** Identify at least three different functions that are supported by the Assert class. (Hint: type in `\Assert`" followed by a period in one of the test classes in the `\ExpediaTest`" project and use the documentation.)

DoesNotThrow  
Inconclusive  
IsNaN

**[Question 7 (4 pts)]** Please explain the functions that you identified briefly.

DoesNotThrow – verifies a delegate does NOT throw an exception  
Inconclusive - throws `NUnit.Framework.InconclusiveException`; causes the test to be reported as Inconclusive  
IsNaN – verifies that the double passed-in is an NaN value

**[Question 8 (4 pts)]** You may have noticed that there are two methods on the assert class named `AreEqual` and `AreSame`. These methods may seem identical, but there is a critical difference between them. Please explain the difference.

`AreEqual` checks for the same value between two objects  
`AreSame` checks for the same reference between two objects

**[Question 9 (4 pts)]** What is the unit test `TestThatHotelInitializes` verifying?

It's verifying the Hotel class initializes correctly

**[Question 10 (4 pts)]** What is the generic algorithm for calculating `getBasePrice`? (You may look at `Hotel.cs`.)

`45 * numberOfNightsToRent`

**[Question 11 (4 pts)]** What cases are tested with these new tests?

The Base Price for a 1, 2, & 10-day stay

**[Question 12 (4 pts)]** Why don't we have to include `Assert.IsNotNull(target)` in each of these tests?

Since it's the first thing we tested and none of the functions being tested modify the target's existence, we do not need to check for a null target again as it would have been detected at the start of testing.

**[Question 13 (4 pts)]** Why does this test expect an exception? (Please look at Hotel.cs.)

The nights to rent must be >0; otherwise, the `ArgumentOutOfRangeException` will be thrown

**[Question 14 (4 pts)]** Please provide the attribute declaration to accompany a test case that expects an `OutOfMemoryException`.

`[ExpectedException(typeof(OutOfMemoryException))]`