Book A Cab

**Grade settings**: Maximum grade: 100  
**Run**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes

[***Click here to download the code skeleton***](https://cognizant.tekstac.com/pluginfile.php/74068/mod_vpl/intro/BookACab.zip)

***Cool Cab***is a famous Cab Service in the city. They have developed an application for taking various reports based on their cab service. The details of the various functions supported by the system are provided in this case study.

You are required to write Junit test case and check the correctness of the application developed.

**Functional Requirements:**

The application has the below classes and methods implemented.

You are provided with a model class BookCab

**Component Specification:**BookCab**(Model Class)**

|  |  |  |
| --- | --- | --- |
| **Type(Class)** | **Attributes** | **Methods** |
| BookCab | **int** bookingId  Date bookDate  String cabType  String chaufferId  String pickupPlace  String dropPlace  int distanceTravelled  double totalFare | Necessary getters and setters are provided.  Parameterized constructor and a no-arg constructor are provided as a part of the code skeleton |

Here the cabType can take a value either “Prime” or “Micro” or “Mini”.

[Note: Values are case insensitive].

**Component Specification:**InvalidBookCabException **(This class inherits the Exception Class)**

|  |  |
| --- | --- |
| **Type(Class)** | **Methods** |
| InvalidBookCabException | Provided with a single argument constructor – InvalidBookCabException(String message) |

You are also provided with a utility class BookCabService and the below business requirements are implemented in it for which JUnit test cases are to be written and tested.

You are also provided with a utility class BookCabService with business methods.

**Component Specification:**BookCabService**(Utility Class)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | **Type(Class)** | **Attributes** | **Methods** |
| BookCabService | BookCabService | List<BookCab> bookCabList | Getter and setter for the bookCabList are provided. |

The below are the requirements  implemented in the Utility class for which JUnit test cases are to be written and tested.

**Component Specification:**BookCabService **(Utility Class)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Type (Class)** | **Methods** | **Responsibilities** | **Exception** |
| Validating the Chauffer Id | BookCabService | public boolean validateChaufferId(String chaufferId) | Validate the Chauffer Id.  If valid return true , else this method should throw a user defined exception | Throw a user defined exception “InvalidBookCabException” if the Chauffer Id does not contain FLY followed by ‘/’ followed by 3 digits, followed by ‘/’ and followed by 4 digits. |
| View Booking Details by Booking Id | BookCabService | public BookCab viewBookingById(int bookingId) | This method should return the BookCab object with the bookingId passed as parameter from bookCabList.  If the bookCabList  is empty or if there is no Cab details  with  the given bookingId it should throw a user defined exception | Throw a user defined exception “InvalidBookCabException” if the bookCabList is empty or if there is no Cab Details in the given Booking Id. |
| View the list of Booking details for a given date of booking | BookCabService | public List<BookCab> viewBookingByBookDate(Date bookDate) | This method takes the date of booking as argument. It should return the list of BookCab Objects for the given date of booking. If the bookCabList is empty it should throw a user defined exception. | Throw a user defined exception “InvalidBookCabException” if the bookCabList is empty. |
| View Total income of each Chauffer | BookCabService | public Map<String,Double> viewTotalIncomeChaufferwise() | This method should return the income of each Chauffer based on Chauffer Id from the bookCabList. It returns a Map with key as Chauffer Id and value as income calculated.  If the bookCabList is empty it should throw a user defined exception. | Throw a user defined exception “InvalidBookCabException” if the bookCabList is empty. |
| View the Booking details based on the Pickup place | BookCabService | public Map<String,List<BookCab>> viewBookingPickupPlacewise() | This method should return the booking details for each Pickup place based on the bookCabList. It returns a Map with key as Pickup place and value as list of booking details.  If the bookCabList is empty it should throw a user defined exception. | Throw a user defined exception “InvalidBookCabException” if the bookCabList is empty. |
| Count Booking based on the Chauffer Id | BookCabService | public int countBookingByChauffer(String chaufferId) | This method takes the chauffer id as argument. It should return the count of bookings for the given chauffer id. If the bookCabList is empty it should throw a user defined exception. | Throw a user defined exception “InvalidBookCabException” if the bookCabList is empty. |

You need to write Junit test for the BookCabService class.

**Testing Scenarios:**

You are provided with a class “BookCabServiceTest” to do this testing.

**Note:**

To perform testing the bookCabList should contain objects of BookCab.

To do this, in BookCabServiceTest class you are provided with a setup method. Use this method to initialize the  bookCabList  attribute in BookCabclass.

Create few objects for BookCabService, populate a list with these objects and set the bookCabList to this list using the setBookCabList method in BookCabclass.

The below are the test methods to be implemented in BookCabServiceTest class.

|  |  |
| --- | --- |
| **Test Method** | **Scenarios / Responsibilities** |
| test11ValidateChaufferId | This method should test the validateChaufferId method when a valid Chauffer Id is passed as parameter |
| test12ValidateChaufferIdWhenInvalid | This method should test the validateChaufferId method when invalid value is passed as parameter  validateChaufferId is expected  to throw InvalidBookCabException when Chauffer Id is invalid.  Write JUnit to test for the exception thrown either by using appropriate annotation or by using try catch block. |
| test13ViewBookingByIdWhenValid | This method should test the correctness of viewBookingById method.  Perform testing for the correctness of the value returned. |
| test14ViewBookingByIdWhenInvalid | This method should test the viewBookingById method when invalid value is passed as parameter  viewBookingById is expected  to throw InvalidBookCabException when Id is invalid.  Write JUnit to test for the exception thrown either by using appropriate annotation or by using try catch block. |
| test15ViewBookingByBookDate | This method should test the correctness of viewBookingByBookDate method.  Perform testing for the correctness of the value returned. |
| test16ViewTotalIncomeChaufferwise | This method should test the correctness of viewTotalIncomeChaufferwise method.  Perform testing for the correctness of the value returned. |
| test17ViewBookingPickupPlacewise | This method should test the correctness of viewBookingPickupPlacewise method.  Perform testing for the correctness of the value returned. |
| test18CountBookingByChauffer | This method should test the correctness of countBookingByChauffer method.  Perform testing for the correctness of the value returned. |
| test19ViewBookingByBookDateForEmptyList | This method should test the viewBookingByBookDate method when invalid value is passed as parameter  viewBookingByBookDate is expected  to throw InvalidBookCabException when Chauffer Id is invalid.  Write JUnit to test for the exception thrown either by using appropriate annotation or by using try catch block. |
| test20ViewTotalIncomeChaufferwiseForEmptyList | This method should test the viewTotalIncomeChaufferwise method when invalid value is passed as parameter  viewTotalIncomeChaufferwise is expected  to throw InvalidBookCabException when Chauffer Id is invalid.  Write JUnit to test for the exception thrown either by using appropriate annotation or by using try catch block. |
| test21ViewBookingPickupPlacewiseForEmptyList | This method should test the correctness of viewBookingPickupPlacewise method for an empty bookCabList.  viewBookingPickupPlacewise method is expected  to throw InvalidBookCabException when bookCabList is empty.  Write JUnit to test for the exception thrown  either by using appropriate annotation or by using try catch block |
| test22CountBookingByChaufferForEmptyList | This method should test the correctness of countBookingByChauffer method for an empty bookCabList.  countBookingByChauffer method is expected  to throw InvalidBookCabException when bookCabList is empty.  Write JUnit to test for the exception thrown  either by using appropriate annotation or by using try catch block |

Implement the test methods and provide the needed annotation to all the methods in BookCabServiceTest class.

Also this class is provided with the annotation, so that the test methods are executed in ascending order of the test method names.

You are provided with a Main class with the main method to check the correctness of the test methods written in BookCabServiceTest class.

Having completed writing the test methods, uncomment the code in Main class and execute the main method.