By Hritaban Ghosh 19CS30053

Test Plan

Unit Tests

Unit Testing Date Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the list of month names in static constant monthNames.
- Check the list of day names in static constant dayNames.
- Check the value of the maximum valid year stored in static constant MAX_VALID_YR.
- Check the value of the minimum valid year stored in static constant MIN_VALID_YR.

Test Scenarios for Checking the Static Member Functions

Consider the static qualified member funcitons of the class. The test scenarios are:

- Check the isLeap utility function.
    1. Check if it returns true upon a leap year input.
    2. Check if it returns false upon non-leap year input.

Test Scenarios for Construction of Objects

Consider the Date(unsigned int d, unsigned int m, unsigned int y) constructor. The test scenario is:

- The data members are correctly initialized. To check if the data members date_ of type unsigned int, month_ of enum type Month and year_ of type unsigned int.

Test Scenarios for Checking the print() member function

Consider the print() member function. The test scenario is:

- Check if the function prints in dd/MMM/yy format. Example: 2/Mar/2021

Test Scenarios for Checking the validDate() member function

Consider the validDate() member function. The test scenario is:

- Check if the function returns true upon inputting a valid date.

P.T.O.

- Check if it returns false upon invalid date input. Example: 29/Feb/2021

Test Scenarios for Checking the day() member function

Consider the day() member function. The test scenario is:

- Check if the function returns the correct day corresponding to the date. Example: “Tuesday” for 2/Mar/2021.

Test Scenarios for equality operator

Consider the overloaded equality operator friend bool operator==(const Date&) const. The test scenarios are :

- Check if the operator returns true if both dates are same.
- Check if it returns false if they are different.

Test Scenarios for inequality operator

Consider the overloaded inequality operator bool operator!=(const Date&) const. The test scenarios are :

- Check if the operator returns false if both dates are same.
- Check if it returns true if they are different.

Test Scenarios for insertion operator

Consider the overloaded insertion operator std::ostream& operator<<(std::ostream&, const Date&). The test scenario is:

- Check if the operator inserts the date to the console in dd/MMM/yy format. Example: 2/Mar/2021

By Hritaban Ghosh 19CS30053

Unit Testing Station Class

Test Scenarios for Construction of Objects

Consider the Station(std::string) constructor. The test scenario is:

- The data members are correctly initialized. To check if the data member name_ of type std::string is correctly initialized.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the name of the station correctly.

Test Scenarios for Checking the GetDistance() member function

Consider the GetDistance() member function. The test scenario is:

- Check if the function returns the distance between itself and another station correctly according to the distance matrix in Railways Class.

Test Scenarios for equality operator

Consider the overloaded equality operator bool operator==(const Station&) const. The test scenarios are :

- Check if the operator returns true if both stations are same.
- Check if it returns false if they are different.

Test Scenarios for inequality operator

Consider the overloaded inequality operator bool operator!=(const Station&) const. The test scenarios are :

- Check if the operator returns false if both stations are same.
- Check if it returns true if they are different.

Test Scenarios for insertion operator

Consider the overloaded insertion operator friend std::ostream& operator<<(std::ostream&, const Station&). The test scenario is:

- Check if the operator inserts the name of the Station to the console.

P.T.O.

By Hritaban Ghosh 19CS30053

Unit Testing Railways Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the list of stations in static constant sStations.
- Check the list of distance matrix in static constant sDistances.

Test Scenarios for Construction of the Singleton Object

Consider the static const Railways& IndianRailways() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the GetDistance() member function

Consider the GetDistance() member function. The test scenario is:

- Check if the function returns the distance between two stations correctly according to the distance matrix.

Test Scenarios for insertion operator

Consider the overloaded insertion operator friend std::ostream& operator<<(std::ostream&, const Railways&). The test scenario is:

- Check if the operator inserts the all the information of the Railways to the console.

By Hritaban Ghosh 19CS30053

Unit Testing AC2Tier Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value stored in the sloadFactor.

Test Scenarios for Construction of the Singleton Object

Consider the static const AC2Tier& Instance() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the IsSitting() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the GetNumberOfTiers() member function

Consider the GetNumberOfTiers() member function. The test scenario is:

- Check if the function returns 2.

Test Scenarios for Checking the GetLoadFactor() member function

Consider the GetLoadFactor() member function. The test scenario is:

- Check if the function returns the same value as sloadFactor.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the “AC 2 Tier”.

Test Scenarios for Checking the IsAC() member function

Consider the IsAC() member function. The test scenario is:

- Check if the function returns true.

Test Scenarios for Checking the IsLuxury() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

P.T.O.

By Hritaban Ghosh 19CS30053

Unit Testing AC3Tier Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value stored in the sloadFactor.

Test Scenarios for Construction of the Singleton Object

Consider the static const AC3Tier& Instance() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the IsSitting() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the GetNumberOfTiers() member function

Consider the GetNumberOfTiers() member function. The test scenario is:

- Check if the function returns 3.

Test Scenarios for Checking the GetLoadFactor() member function

Consider the GetLoadFactor() member function. The test scenario is:

- Check if the function returns the same value as sloadFactor.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the “AC 3 Tier”.

Test Scenarios for Checking the IsAC() member function

Consider the IsAC() member function. The test scenario is:

- Check if the function returns true.

Test Scenarios for Checking the IsLuxury() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

By Hritaban Ghosh 19CS30053

Unit Testing ACChairCar Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value stored in the sloadFactor.

Test Scenarios for Construction of the Singleton Object

Consider the static const ACChairCar& Instance() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the IsSitting() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns true.

Test Scenarios for Checking the GetNumberOfTiers() member function

Consider the GetNumberOfTiers() member function. The test scenario is:

- Check if the function returns 0.

Test Scenarios for Checking the GetLoadFactor() member function

Consider the GetLoadFactor() member function. The test scenario is:

- Check if the function returns the same value as sloadFactor.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the “AC Chair Car”.

Test Scenarios for Checking the IsAC() member function

Consider the IsAC() member function. The test scenario is:

- Check if the function returns true.

Test Scenarios for Checking the IsLuxury() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

P.T.O.

By Hritaban Ghosh 19CS30053

Unit Testing ACFirstClass Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value stored in the sloadFactor.

Test Scenarios for Construction of the Singleton Object

Consider the static const ACFirstClass& Instance() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the IsSitting() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the GetNumberOfTiers() member function

Consider the GetNumberOfTiers() member function. The test scenario is:

- Check if the function returns 2.

Test Scenarios for Checking the GetLoadFactor() member function

Consider the GetLoadFactor() member function. The test scenario is:

- Check if the function returns the same value as sloadFactor.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the “AC First Class”.

Test Scenarios for Checking the IsAC() member function

Consider the IsAC() member function. The test scenario is:

- Check if the function returns true.

Test Scenarios for Checking the IsLuxury() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns true.

By Hritaban Ghosh 19CS30053

Unit Testing FirstClass Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value stored in the sloadFactor.

Test Scenarios for Construction of the Singleton Object

Consider the static const FirstClass& Instance() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the IsSitting() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the GetNumberOfTiers() member function

Consider the GetNumberOfTiers() member function. The test scenario is:

- Check if the function returns 2.

Test Scenarios for Checking the GetLoadFactor() member function

Consider the GetLoadFactor() member function. The test scenario is:

- Check if the function returns the same value as sloadFactor.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the “First Class”.

Test Scenarios for Checking the IsAC() member function

Consider the IsAC() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the IsLuxury() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

P.T.O.

By Hritaban Ghosh 19CS30053

Unit Testing SecondSitting Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value stored in the sloadFactor.

Test Scenarios for Construction of the Singleton Object

Consider the static const SecondSitting& Instance() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the IsSitting() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns true.

Test Scenarios for Checking the GetNumberOfTiers() member function

Consider the GetNumberOfTiers() member function. The test scenario is:

- Check if the function returns 0.

Test Scenarios for Checking the GetLoadFactor() member function

Consider the GetLoadFactor() member function. The test scenario is:

- Check if the function returns the same value as sloadFactor.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the “Second Sitting”.

Test Scenarios for Checking the IsAC() member function

Consider the IsAC() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the IsLuxury() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

By Hritaban Ghosh 19CS30053

Unit Testing Sleeper Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value stored in the sloadFactor.

Test Scenarios for Construction of the Singleton Object

Consider the static const Sleeper& Instance() instance creator. The test scenario is:

- Check whether multiple calls to the instance creator results in obtaining a singleton object.

Test Scenarios for Checking the IsSitting() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the GetNumberOfTiers() member function

Consider the GetNumberOfTiers() member function. The test scenario is:

- Check if the function returns 3.

Test Scenarios for Checking the GetLoadFactor() member function

Consider the GetLoadFactor() member function. The test scenario is:

- Check if the function returns the same value as sloadFactor.

Test Scenarios for Checking the GetName() member function

Consider the GetName() member function. The test scenario is:

- Check if the function returns the “Sleeper”.

Test Scenarios for Checking the IsAC() member function

Consider the IsAC() member function. The test scenario is:

- Check if the function returns false.

Test Scenarios for Checking the IsLuxury() member function

Consider the IsSitting() member function. The test scenario is:

- Check if the function returns false.

P.T.O.

By Hritaban Ghosh 19CS30053

Unit Testing Booking Class

Test Scenarios for Checking the Static Data Members

Consider the static qualified members of the class. The test scenarios are:

- Check the value of the Bare Fare charged per Kilometer stored in static constant sBarePerKM.
- Check the value of the AC Surcharge levied stored in static constant sACSurcharge.
- Check the value of the Luxury Tax Percentage levied stored in static constant sLuxuryTaxPercent.

Test Scenarios for Construction of Objects

Consider the Booking(Station, Station, Date, const BookingClass&, Passenger* passenger_val=NULL) constructor. The test scenario is:

- The data members are correctly initialized. To check if the data members fromStation, toStation, date and bookingClass are correctly initialized.

Test Scenarios for Checking the ComputeFare() member function

Consider the ComputeFare() member function. The test scenario is:

- Check if the function computes the fare correctly.

Test Scenarios for equality operator

Consider the overloaded equality operator bool operator==(const Booking&) const. The test scenarios are :

- Check if the operator returns true if both bookings have the same PNR are same.
- Check if it returns false if they have different PNRs.

Test Scenarios for inequality operator

Consider the overloaded inequality operator bool operator!=(const Booking&) const. The test scenarios are :

- Check if the operator returns false if both bookings have the same PNR are same.
- Check if it returns true if they have different PNRs.

By Hritaban Ghosh 19CS30053

Test Scenarios for insertion operator

Consider the overloaded insertion operator friend std::ostream& operator<<(std::ostream&, const Booking&). The test scenario is:

- Check if the operator inserts all the information of the booking to the console.

Application Tests

Application Test Plan 1:

Test Application For Booking:

This is the Test Application provided in the Assignment. This ApplicationTest1 tests the Railway Booking System for a few scenarios.
This includes only 8 bookings with test output for verification.

Test Output:

```
Application Test 1:

BOOKING SUCCEEDED:
PNR Number : 1
From Station : Mumbai
To Station : Delhi
Travel Date : 15/Feb/2021
Travel Class : AC First Class
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 2
--> Luxury : Yes
Fare : 2776


BOOKING SUCCEEDED:
PNR Number : 2
From Station : Kolkata
To Station : Delhi
Travel Date : 5/Mar/2021
Travel Class : AC 2 Tier
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 2
--> Luxury : No
Fare : 1522
```

By Hritaban Ghosh 19CS30053

```
BOOKING SUCCEEDED:
PNR Number : 3
From Station : Mumbai
To Station : Kolkata
Travel Date : 17/Mar/2021
Travel Class : First Class
--> Mode : Sleeping
--> Comfort : Non-AC
--> Bunks : 2
--> Luxury : No
Fare : 2014


BOOKING SUCCEEDED:
PNR Number : 4
From Station : Mumbai
To Station : Delhi
Travel Date : 23/Mar/2021
Travel Class : AC 3 Tier
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 3
--> Luxury : No
Fare : 1316


BOOKING SUCCEEDED:
PNR Number : 5
From Station : Chennai
To Station : Delhi
Travel Date : 25/Apr/2021
Travel Class : AC Chair Car
--> Mode : Sitting
--> Comfort : AC
--> Bunks : 0
--> Luxury : No
Fare : 1413


BOOKING SUCCEEDED:
PNR Number : 6
From Station : Chennai
To Station : Kolkata
Travel Date : 7/May/2021
Travel Class : Sleeper
--> Mode : Sleeping
--> Comfort : Non-AC
--> Bunks : 3
--> Luxury : No
Fare : 830
```

```
BOOKING SUCCEEDED:
PNR Number : 7
From Station : Mumbai
To Station : Delhi
Travel Date : 19/May/2021
Travel Class : Second Sitting
--> Mode : Sitting
--> Comfort : Non-AC
--> Bunks : 0
--> Luxury : No
Fare : 362


BOOKING SUCCEEDED:
PNR Number : 8
From Station : Delhi
To Station : Mumbai
Travel Date : 22/May/2021
Travel Class : Second Sitting
--> Mode : Sitting
--> Comfort : Non-AC
--> Bunks : 0
--> Luxury : No
Fare : 362
```

My Application Test Plan:

Test Application For Booking:

This is the Test Application which is written by me. It has 14 different cases for tests and it makes use of all Booking Classes and Stations. Thus it thoroughly taps into every aspect of the program. The test output to be matched is given below.

Test Output:

```
My Application Test :

BOOKING SUCCEEDED:
PNR Number : 1
From Station : Mumbai
To Station : Delhi
Travel Date : 15/Feb/2021
Travel Class : AC First Class
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 2
--> Luxury : Yes
```

```
Fare : 2776


BOOKING SUCCEEDED:
PNR Number : 2
From Station : Bangalore
To Station : Kolkata
Travel Date : 5/Mar/2021
Travel Class : AC First Class
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 2
--> Luxury : Yes
Fare : 3571


BOOKING SUCCEEDED:
PNR Number : 3
From Station : Chennai
To Station : Mumbai
Travel Date : 17/Mar/2021
Travel Class : First Class
--> Mode : Sleeping
--> Comfort : Non-AC
--> Bunks : 2
--> Luxury : No
Fare : 1338


BOOKING SUCCEEDED:
PNR Number : 4
From Station : Delhi
To Station : Bangalore
Travel Date : 23/Mar/2021
Travel Class : First Class
--> Mode : Sleeping
--> Comfort : Non-AC
--> Bunks : 2
--> Luxury : No
Fare : 2150


BOOKING SUCCEEDED:
PNR Number : 5
From Station : Kolkata
To Station : Chennai
Travel Date : 25/Apr/2021
Travel Class : AC Chair Car
--> Mode : Sitting
--> Comfort : AC
--> Bunks : 0
--> Luxury : No
Fare : 1087
```

```
BOOKING SUCCEEDED:
PNR Number : 6
From Station : Mumbai
To Station : Delhi
Travel Date : 7/May/2021
Travel Class : AC Chair Car
--> Mode : Sitting
--> Comfort : AC
--> Bunks : 0
--> Luxury : No
Fare : 954


BOOKING SUCCEEDED:
PNR Number : 7
From Station : Bangalore
To Station : Kolkata
Travel Date : 19/May/2021
Travel Class : Second Sitting
--> Mode : Sitting
--> Comfort : Non-AC
--> Bunks : 0
--> Luxury : No
Fare : 468


BOOKING SUCCEEDED:
PNR Number : 8
From Station : Chennai
To Station : Mumbai
Travel Date : 22/May/2021
Travel Class : Second Sitting
--> Mode : Sitting
--> Comfort : Non-AC
--> Bunks : 0
--> Luxury : No
Fare : 335


BOOKING SUCCEEDED:
PNR Number : 9
From Station : Delhi
To Station : Bangalore
Travel Date : 15/Feb/2021
Travel Class : AC 2 Tier
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 2
--> Luxury : No
Fare : 2200


BOOKING SUCCEEDED:
```

```
PNR Number : 10
From Station : Kolkata
To Station : Chennai
Travel Date : 5/Mar/2021
Travel Class : AC 2 Tier
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 2
--> Luxury : No
Fare : 1709


BOOKING SUCCEEDED:
PNR Number : 11
From Station : Mumbai
To Station : Kolkata
Travel Date : 17/Mar/2021
Travel Class : AC 3 Tier
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 3
--> Luxury : No
Fare : 1812


BOOKING SUCCEEDED:
PNR Number : 12
From Station : Chennai
To Station : Delhi
Travel Date : 23/Mar/2021
Travel Class : AC 3 Tier
--> Mode : Sleeping
--> Comfort : AC
--> Bunks : 3
--> Luxury : No
Fare : 1958


BOOKING SUCCEEDED:
PNR Number : 13
From Station : Bangalore
To Station : Mumbai
Travel Date : 25/Apr/2021
Travel Class : Sleeper
--> Mode : Sleeping
--> Comfort : Non-AC
--> Bunks : 3
--> Luxury : No
Fare : 491


BOOKING SUCCEEDED:
PNR Number : 14
From Station : Bangalore
```

```
To Station : Kolkata
Travel Date : 7/May/2021
Travel Class : Sleeper
--> Mode : Sleeping
--> Comfort : Non-AC
--> Bunks : 3
--> Luxury : No
Fare : 936
```