

TEST REPORT

-Ishan Goel(19CS30052)

Unit Testing Plan:

Class Railways:

- Validity is checked.
- The Methods likeGetDistance are tested.
- The friend function overloaded << operator is tested.
- Every output is compared to its Golden Value.

```

=====Testing Railways Class=====

=== Testing operator << ===
Indian Railways
List of all Stations
Mumbai
Delhi
Bangalore
Kolkata
Chennai

Distance between stations
Bangalore -> Chennai = 350
Bangalore -> Delhi = 2150
Bangalore -> Kolkata = 1871
Bangalore -> Mumbai = 981
Chennai -> Bangalore = 350
Chennai -> Delhi = 2180
Chennai -> Kolkata = 1659
Chennai -> Mumbai = 1338
Delhi -> Bangalore = 2150
Delhi -> Chennai = 2180
Delhi -> Kolkata = 1472
Delhi -> Mumbai = 1447
Kolkata -> Bangalore = 1871
Kolkata -> Chennai = 1659
Kolkata -> Delhi = 1472
Kolkata -> Mumbai = 2014
Mumbai -> Bangalore = 981
Mumbai -> Chennai = 1338
Mumbai -> Delhi = 1447
Mumbai -> Kolkata = 2014

===Testing GetDistance() Method ===
Distance between Mumbai and Delhi: 1447

```

Class Station:

- First Construction is tested.
- The Methods like GetName, GetDistance are tested.
- The friend function overloaded << operator is tested.
- Every output is compared to its Golden Value.

```

=====Testing Station Class=====

=== Testing Construction ===
Address of Constructed Object: 0x7ffd1fa9de10
=== Testing GetName() Method ===
Name: Kolkata
=== Testing operator << ===
Kolkata Mumbai
=== Testing GetDistance() Method ===
Distance= 2014
Distance between Chennai and Kolkata = 1659

```

Class Date:

- First Construction and Validity is tested.
- The Methods like getDay, getMonth, getYear, ComputeAge are tested.
- Exception Handling is tested
- The friend function overloaded << operator is tested.
- Every output is compared to its Golden Value.

```
=====Testing Date Class=====
=== Testing Construction ===
Address of Constructed Object: 0x7ffd1fa9de80
=== Testing operator << ===
28/2/2001
=== Testing exception ===
Bad Date
29/2/2021
```

Class Passenger:

- Good and bad passenger objects are constructed.
- We can see that the good passenger gets printed (using << overloaded operator) whereas the bad passenger gives an error message also

```
=====Testing Passenger Class=====
=== Testing Constructor ===
ISHAN Abd
: Date of Birth: 28/2/2001
: Gender: Male
: Aadhar: d8d73g28h221
: Mobile: 7602045111
: Disability Type: 1ow1

=== Testing Exceptions ===
Invalid Aadhar
ISHAN Abd
: Date of Birth: 28/2/2001
: Gender: Male
: Aadhar:
: Mobile: 7602045111
: Disability Type: 1ow1
```

Class BookingClass and Hierarchy:

- Testing << operator. This operator calls methods like GetName, IsSitting, IsAC, GetNumberOfTiers, and IsLuxury methods
- if the printed values match golden values, it works fine.

- Testing virtual GetLoadFactor Method: calling GetLoadFactor of every final concrete class and comparing them to golden values. If no error prints on screen, the method works fine
- Every inherited class object is called and printed.

```

=====Testing Booking Classes Class=====

=== Testing operator << for every Booking Class ===

ACFirstClass
: Fare Load Factor: 6.5
: Mode: Sleeping
: Comfort: AC
: Bunks: 2
: Luxury: Yes
: Reservation Charge: 60
: Min Tatkal Charge : 400
: Max Tatkal Charge : 500
: Min Tatkal Distance : 500

AC2Tier
: Fare Load Factor: 4
: Mode: Sleeping
: Comfort: AC
: Bunks: 2
: Luxury: No
: Reservation Charge: 50
: Min Tatkal Charge : 400
: Max Tatkal Charge : 500
: Min Tatkal Distance : 500

AC3Tier
: Fare Load Factor: 2.5
: Mode: Sleeping
: Comfort: AC
: Bunks: 3
: Luxury: No
: Reservation Charge: 40
: Min Tatkal Charge : 300
: Max Tatkal Charge : 400
: Min Tatkal Distance : 500

```

```

SecondSitting
: Fare Load Factor: 0.6
: Mode: Sitting
: Comfort: Non-AC
: Bunks: 0
: Luxury: No
: Reservation Charge: 15
: Min Tatkal Charge : 10
: Max Tatkal Charge : 15
: Min Tatkal Distance : 100

 Sleeper
: Fare Load Factor: 1
: Mode: Sleeping
: Comfort: Non-AC
: Bunks: 3
: Luxury: No
: Reservation Charge: 20
: Min Tatkal Charge : 100
: Max Tatkal Charge : 200
: Min Tatkal Distance : 500

Fare Loaded Factor of ACFirstClass from Virtual GetLoadFactor() Method = 6.5
Fare Loaded Factor of AC2Tier from Virtual GetLoadFactor() Method = 4
Fare Loaded Factor of AC3Tier from Virtual GetLoadFactor() Method = 2.5
Fare Loaded Factor of ACChairCar from Virtual GetLoadFactor() Method = 2
Fare Loaded Factor of ExecutiveChairCar from Virtual GetLoadFactor() Method = 5
Fare Loaded Factor of FirstClass from Virtual GetLoadFactor() Method = 3
Fare Loaded Factor of SecondSitting from Virtual GetLoadFactor() Method = 0.6
Fare Loaded Factor of Sleeper from Virtual GetLoadFactor() Method = 1

```

Class BookingCategory and Hierarchy(Categories) :

- The whole hierarchy consists of singleton classes.
- Unit testing checking whether every singleton is properly implemented or not.
- Each instance is constructed and the address is printed

```

=====Testing Category Class=====

=====Testing Construction of every Class=====
Constructing General Object
Address of General Category : 0x5574d0fa5178
=====
Constructing Tatkal Object
Address of Tatkal Category : 0x5574d0fa5188
=====
Constructing PremiumTatkal Object
Address of PremiumTatkal Category : 0x5574d0fa5198
=====
Constructing SeniorCitizen Object
Address of SeniorCitizen Category : 0x5574d0fa51a8
=====
Constructing Blind Object
Address of Blind Category : 0x5574d0fa51b8
=====
Constructing OH Object
Address of OH Category : 0x5574d0fa51c8
=====
Constructing TB Object
Address of TB Category : 0x5574d0fa51d8
=====
Constructing Cancer Object
Address of Cancer Category : 0x5574d0fa51e8
=====

```

Class Booking:

- Testing of this class is in application testing

Application Testing Report

Test Suite 1:

Approach - All test cases with "**positive outputs**" and of every category are covered

Test Cases -

- Test Case 1 -

Condition covered - Construction of booking of General category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then General class booking is created and its output is considered for evaluation.

PASSED

```
BOOKING SUCCEEDED:
PNR Number = 1
From Station = Mumbai
To Station = Kolkata
Reservation Date = 9/4/2021
Travel Date = 20/4/2021
Travel Class = AC3Tier
: Fare Load Factor: 2.5
: Mode: Sleeping
: Comfort: AC
: Bunks: 3
: Luxury: No
: Reservation Charge: 40
: Min Tatkal Charge : 300
: Max Tatkal Charge : 400
: Min Tatkal Distance : 500
Fare = 2558
```

- Test Case 2 -

Condition covered - Construction of booking of Tatkal category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Tatkal class booking is created and its output is considered for evaluation.

PASSED

```
BOOKING SUCCEEDED:
PNR Number = 2
From Station = Bangalore
To Station = Kolkata
Reservation Date = 9/4/2021
Travel Date = 10/4/2021
Travel Class = ACFirstClass
: Fare Load Factor: 6.5
: Mode: Sleeping
: Comfort: AC
: Bunks: 2
: Luxury: Yes
: Reservation Charge: 60
: Min Tatkal Charge : 400
: Max Tatkal Charge : 500
: Min Tatkal Distance : 500
Fare = 6641
```

- Test Case 3 -

Condition covered - Construction of booking of Premium Tatkal category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Premium Tatkal class booking is created and its output is considered for evaluation.

PASSED


```
BOOKING SUCCEEDED:
PNR Number = 3
From Station = Bangalore
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 20/5/2021
Travel Class = ACChairCar
: Fare Load Factor: 2
: Mode: Sitting
: Comfort: AC
: Bunks: 0
: Luxury: No
: Reservation Charge: 40
: Min Tatkal Charge : 125
: Max Tatkal Charge : 225
: Min Tatkal Distance : 250
Fare = 640
```

- Test Case 4 -

Condition covered - Construction of booking of Senior Citizen category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Senior Citizen class booking is created and it's output is considered for evaluation.

PASSED

```
BOOKING SUCCEEDED:
PNR Number = 4
From Station = Delhi
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = Sleeper
: Fare Load Factor: 1
: Mode: Sleeping
: Comfort: Non-AC
: Bunks: 3
: Luxury: No
: Reservation Charge: 20
: Min Tatkal Charge : 100
: Max Tatkal Charge : 200
: Min Tatkal Distance : 500
Fare = 565
```

- Test Case 5 -

Condition covered - Construction of booking of Blind category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Blind class booking is created and its output is considered for evaluation.

PASSED

```
BOOKING SUCCEEDED:
PNR Number = 5
From Station = Kolkata
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 4/6/2021
Travel Class = SecondSitting
: Fare Load Factor: 0.6
: Mode: Sitting
: Comfort: Non-AC
: Bunks: 0
: Luxury: No
: Reservation Charge: 15
: Min Tatkal Charge : 10
: Max Tatkal Charge : 15
: Min Tatkal Distance : 100
Fare = 139
```

- Test Case 6 -

Condition covered - Construction of booking of Orthopaedically Handicapped category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Orthopaedically Handicapped class booking is created and its output is considered for evaluation.

PASSED

```
BOOKING SUCCEEDED:
PNR Number = 6
From Station = Delhi
To Station = Bangalore
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = ExecutiveChairCar
: Fare Load Factor: 5
: Mode: Sitting
: Comfort: AC
: Bunks: 0
: Luxury: Yes
: Reservation Charge: 60
: Min Tatkal Charge : 400
: Max Tatkal Charge : 500
: Min Tatkal Distance : 250
Fare = 1404
```

- Test Case 7 -

Condition covered - Construction of booking of Cancer category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Cancer class booking is created and its output is considered for evaluation.

PASSED

```
BOOKING SUCCEEDED:
PNR Number = 7
From Station = Delhi
To Station = Mumbai
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = AC3Tier
: Fare Load Factor: 2.5
: Mode: Sleeping
: Comfort: AC
: Bunks: 3
: Luxury: No
: Reservation Charge: 40
: Min Tatkal Charge : 300
: Max Tatkal Charge : 400
: Min Tatkal Distance : 500
Fare = 40
```

- Test Case 8 -

Condition covered - Construction of booking of TB category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then TB class booking is created and its output is considered for evaluation.

PASSED

```
BOOKING SUCCEEDED:
PNR Number = 8
From Station = Kolkata
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = FirstClass
: Fare Load Factor: 3
: Mode: Sleeping
: Comfort: Non-AC
: Bunks: 2
: Luxury: Yes
: Reservation Charge: 50
: Min Tatkal Charge : 400
: Max Tatkal Charge : 500
: Min Tatkal Distance : 500
Fare = 672
```

Test Suite 2 :-

Approach - All test cases with "**negative outputs**" and of every exceptions are covered

Test Cases -

- Test Case 1 -

Condition covered - Construction of booking with invalid date

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong booking date and its output is considered for

evaluation.

PASSED

```
Bad Date  
Bad Booking Date
```

- Test Case 2 -

Condition covered - Construction of booking with past date

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong booking date and its output is considered for

PASSED

```
Bad Booking Date
```

- Test Case 3 -

Condition covered - Construction of booking with invalid aadhar, mobile number and name

Step Involved - Construct a passenger object using its constructor with wrong inputs. Then instance of class booking is created and its output is considered for

PASSED

```
Invalid Name  
Invalid Mobile  
Invalid Aadhar
```

- Test Case 4 -

Condition covered - Construction of booking with invalid age for senior citizen

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong age and its output is considered for

PASSED

Senior citizen Error

- Test Case 5 -

Condition covered - Construction of booking with wrong disability

Step Involved - Construct a passenger object using its constructor with wrong disability. Then instance of class booking is created and its output is considered for

PASSED

Divyaang Error

- Test Case 6 -

Condition covered - Construction of booking with wrong category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong category and its output is considered for

PASSED

Category Error

- Test Case 7 -

Condition covered - Construction of booking with wrong station

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong station and its output is considered for

PASSED

Same Station Invalid

The End