Station Class ;
    private data members:
        name_ => this will store the name of the station, string
    public methods:
        GetDistance() => this will return the distance between the two stations
        GetName() => This will return the name of the station, const as the Station object might be const
        operator==() is overloaded because equality of two stations will be used
        friend function for out-streaming

Date Class :
    private data members:
        date_ => this will store the current date, unsigned integer
        month_ => this will store the current month, Month type
        year_ => this will store the current year, unsigned integer
        Month, Day => enumerates the month and day names, respectively, to numbers, enum type
    public data members:
        constructor with arguments => date, month, year
        copy assignment operator
        copy constructor
        CheckLeapYear() => this will check if the year is leap or not, boolean
        CheckSpanOfYear() => this will check if the difference between the current and argument is more than one year, boolean
        CalculateAge() => this will calculate the difference from argument and current date, integer
        Print() => printing the date in dd/mm/yyyy format, const as date object might be const
    static constants declared as public:
        dayNames string array to store day names as string
        monthNames string array to store month names as string
    static methods declared as public:
        UnitTestDate() will test the functionalities of the Date Class

Railways Class : This will be a singleton object
    private:
        constructor is made private in order to make a singleton object
        pointer to the same class in order to have a singleton object
    public:
        copy constructor and copy assignment operator are blocked
        GetDistance() => this will return the distance between the two stations

static:
sDistStations => this will store the pair of stations and their distances in form of a vector of pair of pair of stations and integer
sStations => this will store the list of stations in form of vector

BookingClasses Class: This is an abstract class
protected data members:
reservationCharges_ => this will store the reservation charges of the class, float
name => this will store the name of the booking class, string
loadFactor => this will store the loading factor of the class, float
public functions:
All functions are pure virtual functions and const since a const object will be calling these functions
GetName() => this will return the name of the class, string
GetNumberOfTiers() => this will return the number of tiers in the booking class, integer
IsAC() => this will return true if the booking class is AC or else false, boolean
IsSitting() => this will return true if the booking class is sitting or else false, boolean
IsLuxury() => this will return true if the booking class is Luxury or else false, boolean
GetLoadFactor() => this will return the load factor of the class, float
GetMaximumTatkalCharge() => this will return the maximum tatkal charge, integer
GetMinimumTatkalDistance() => this will return the minimum tatkal distance, integer
GetMinimumTatkalCharge() => this will return the minimum tatkal charge, integer
GetTatkalLoadFactor() => this will return the tatkal load factor of the booking class, float

AC2Tier Class: This class is derived from the BookingClasses Class
private:
numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked

Type() => this will return the singleton Object of the class
friend function for output streaming
static function for unit testing

FirstClass Class: This class is derived from the BookingClasses Class
private:
numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked
Type() => this will return the singleton Object of the class
friend function for output streaming
static function for unit testing

ACFirstClass Class: This class is derived from the BookingClasses Class
private:
numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked
Type() => this will return the singleton Object of the class
friend function for output streaming

static function for unit testing

AC3Tier Class: This class is derived from the BookingClasses Class
private:
numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked
Type() => this will return the singleton Object of the class
friend function for output streaming
static function for unit testing

Sleeper Class: This class is derived from the BookingClasses Class
private:
numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked
Type() => this will return the singleton Object of the class
friend function for output streaming
static function for unit testing

SecondSitting Class: This class is derived from the BookingClasses Class
private:
numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked
Type() => this will return the singleton Object of the class
friend function for output streaming
static function for unit testing

ExecutiveChairCar Class: This class is derived from the BookingClasses Class
private:
numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked
Type() => this will return the singleton Object of the class
friend function for output streaming
static function for unit testing

ACChairCar Class: This class is derived from the BookingClasses Class
private:

numberOfTiers_ => this will store the number of tiers, integer, const since it won't change in future
isAC_ => this is a boolean value that will store if the Class is AC or not, const since it won't change in the future
hasSeat_ => this is a boolean value that will store whether the Class has seats or not, const since it won't change in the future
isLuxury_ => this is a boolean value that will store if the Class is a luxury or not
maximumTatkalCharge_ => this will store the maximum tatkal charge, integer
tatkalLoadFactor_ => this will store the tatkal load factor of the class, float
minimumTatkalDistance_ => this will store the minimum tatkal distance, integer
minimumTatkalCharge_ => this will store the minimum tatkal charge, integer
static pointer to the singleton object of the class
constructor is private to make it a singleton class

public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment are blocked
Type() => this will return the singleton Object of the class
friend function for output streaming
static function for unit testing

BookingCategory Class: this is an abstract base class

protected data members:
name_ => this will store the name of the booking class, string

public functions:
CreateBooking() => this method takes booking class type, stations, dates as arguments and return a pointer to the booking object cerated, Booking *
GetName() => this will return the name of the booking class, string

General Class: This class is derived from BookingCategory Class

private:
constructor is private to make the class singleton

public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming
static function for unit testing

Tatkal Class: This class is derived from BookingCategory Class

private:
constructor is private to make the class singleton

public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming

static function for unit testing

PremiumTatkal Class: This class is derived from BookingCategory Class
private:
constructor is private to make the class singleton
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming
static function for unit testing

Concessions Class: This is derived from the BookingCategory, this is and abstract base class for the classes which are for concessions

SeniorCitizen Class: derived from the Concessions Class
private:
constructor is made private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming
static function for unit testing

Ladies Class: derived from the Concessions Class
private:
constructor is made private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming
static function for unit testing

Divyaang Class: Derived from the Concessions class, it is an abstract base class for those classes which are under Divyaang

Blind Class: derived from Divyaang Class
private:
constructor is made private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming

static function for unit testing
static public constants:
sConcessions => this store the master data of the class in the form of vector of pair of string and float value

OrthopedicallyHandicapped Class: derived from Divyaang Class
private:
constructor is made private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming
static function for unit testing
static public constants:
sConcessions => this store the master data of the class in the form of vector of pair of string and float value

TBPatient Class: derived from Divyaang Class
private:
constructor is made private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming
static function for unit testing
static public constants:
sConcessions => this store the master data of the class in the form of vector of pair of string and float value

CancerPatient Class: derived from Divyaang Class
private:
constructor is made private to make it a singleton class
public:
pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this function will return the singleton object of the class
friend function for output streaming
static function for unit testing
static public constants:
sConcessions => this store the master data of the class in the form of vector of pair of string and float value

Booking class: this is an abstract base class
protected data members:

pnr_ => this will store the PNR of the current booking as an integer
fromStation_ => this will store the start station of the booking, Station
toStation_ => this will store the destination station of the booking, Station
dateOfReservation_ => this will store the date of reservation, Date
dateOfBooking_ => this will store the date of booking, Date
bookingStatus_ => this will store whether the booking was successful or not, boolean
bookingMessage_ => will store the message for booking as a string
passenger_ => will store the passenger details when the booking is initiated

public methods:

ComputeFare() => This will compute the fare, virtual

static public constants:

sBaseFarePerKM => this will store the float value of the base fare, float
sBookings => this will store all the bookings done as a vector of booking pointers
sBookingPNRSerial => this will store the unique PNR Id as an integer, integer

GeneralBooking Class: Derived from the Booking Class

public methods:

ComputeFare() => This is a concrete method to compute fare according to GeneralBooking class

PremiumTatkal Class: Derived from the Booking Class

public methods:

ComputeFare() => This is a concrete method to compute fare according to PremiumTatkal class

Tatkal Class: Derived from the Booking Class

public methods:

ComputeFare() => This is a concrete method to compute fare according to Tatkal class

Concessions Class: Derived from the Booking Class, this is an abstract class for the classes which are for concessions

Ladies Class: Derived from the Concessions Class

public methods:

ComputeFare() => This is a concrete method to compute fare according to Ladies class

SeniorCitizen Class: Derived from the Concessions Class

public methods:

ComputeFare() => This is a concrete method to compute fare according to SeniorCitizen class

Divyaang Class: Derived from the Concessions class, it is an abstract base class for

those classes which are under Divyaang

Blind Class: Derived from the Divyaang Class
    public methods:
        ComputeFare() => This is a concrete method to compute fare according to SeniorCitizen class

OrthopedicallyHandicapped Class: Derived from the Divyaang Class
    public methods:
        ComputeFare() => This is a concrete method to compute fare according to SeniorCitizen class

CancerPatient Class: Derived from the Divyaang Class
    public methods:
        ComputeFare() => This is a concrete method to compute fare according to SeniorCitizen class

TBPatient Class: Derived from the Divyaang Class
    public methods:
        ComputeFare() => This is a concrete method to compute fare according to SeniorCitizen class

Gender Class: This is a abstract base class
    protected:
        name_ => this will store the name of the gender, string
    public:
        GetTitle() => this will return the title for the current object, string
        IsMale() => this will return true if the person is male or else false, boolean
        GetName() => this will return the name of the current gender, string

Male Class: Derived from the Gender Class
    private:
        constructor is made private to make the class singleton
        static pointer to the singleton object of the class
    public:
        pure virtual functions from the abstract class are implemented as given above
        copy constructor and copy assignment operator are blocked
        Type() => this will return the singleton object of the class
        friend function for output streaming
        static function for unit testing

Female Class: Derived from the Gender Class
    private:
        constructor is made private to make the class singleton
        static pointer to the singleton object of the class
    public:

pure virtual functions from the abstract class are implemented as given above
copy constructor and copy assignment operator are blocked
Type() => this will return the singleton object of the class
friend function for output streaming
static function for unit testing

Passenger Class:

private data members:
firstName_ => this will store the first name of the passenger, string
middleName_ => this will store the middle name of the passenger (optional), string
lastName_ => this will store the last name of the passenger, string
dateOfBirth_ => this will store the date of birth of the passenger, Date
gender_ => this will store the gender of the passenger, Gender
aadhar_ => this will store the aadhar of the passenger, string
mobile_ => this will store the mobile of the passenger, string
disabilityType_ => this will store the disability type of the passenger (optional), string
disabilityId_ => this will store the disability id of the passenger (optional), string
public methods:
CheckAadhar() => this will validate the aadhar of the passenger, boolean
GetAadhar() => this will return the aadhar of the passenger, string
CheckMobile() => this will validate the mobile number of the passenger, boolean
GetName() => this will return the name of the passenger, string
CheckName() => this will validate the name of the passenger, boolean
GetMobile() => this will return the mobile number of the passenger, string
GetGender() => this will return the gender of the passenger, Gender
CheckDateOfBirth() => this will validate the date of birth of the passenger, boolean
GetDisabilityType() => this will return the disability type of the passenger, string
GetDisabilityId() => this will return the disability ID of the passenger, string