Airline Booking System

Sai vikram Karthikeyan

The University of Texas at Dallas

MIS 6382: Object Oriented Programming in Python

Professor Vatsal Maru

December 21, 2025

OOP Project: Airline Booking System

Abstract:

This project “Airline Booking system” is developed based on the Python and Object Oriented Programming(OOP) concepts. In this the requirements that helps to build an airline booking system that manages are: Different flight types Domestic and International, Passenger types (Adult, Child, Senior), Giving loyalty membership program, and Finally, giving booking operation with different dynamic pricing. This project going to use OOPS concept like Encapsulation, Polymorphism, Inheritance these techniques going to help in creating this project. This will help in code reusability, Data redundancy, protected, makes data private, Robust. These things help in creating a complete system where inherits multiple classes and that interact with each other to perform the particular task .

Project Overview:

This Airline Booking System contains four part each part has its own responsibility and purpose to build this project.

- Part 1: Different flight types (Domestic and International)
- Part 2: Different passenger types (Adult, Child, Senior)
- Part 3: A Loyalty membership program based on miles.
- Part 4: Booking Operation System with dynamic pricing.

Objectives:

Encapsulation:

The main purpose of Encapsulation is to protect sensitive data for example where the user cannot see the process or method happening only the person who created the code can know that and for this Double underscore is way that know that particular attributes is encapsulated.

For example in this project the encapsulated parts are:

Part 1: Private (Encapsulated): In Flight class __base_price, __available_economy, __available_business are private attributes.

Part 3: Private: In Loyalty Member __miles tier (Silver/Gold/Platinum) are private attributes

Part 4: Private: Booking __total_cost are private attributes

Polymorphism:

In this project Polymorphism is used in many places. where the main features of this polymorphism is having the capable of define in different forms . which means the methodology of one parent class can be taken and modify that and use in different sub class and this polymorphism has two types of methods that are used : overloading and overriding.

For example in this project the Polymorphism parts are:

Part 1: The Polymorphism are used for Domestic and International in calculating price for adding the state tax.

Part 2: The Polymorphism are used in Passenger class for discounts and baggage allowances.

All Parts: __repr__(self) - This String representation is example of Override Polymorphism

Inheritance:

The inheritance is basically can able to inherit the method from parent to child class or either

way. The main thing is to help in reusability of code . There are types of inheritance we are used in this project are single inheritance, Multi-level inheritance, Mutiple inheritance.

For example in this project the inheritance parts are:

Part 1: Single inheritance: Domestic Flight and International Flight inherits from Flight class.

Part 2: Multi-level inheritance: Adult passenger , Child passenger, Senior passenger these three level inherits from Passenger class.

Part 3:Multiple inheritance : Passenger + LoyaltyMember these are Multiple inheritance as they are using two base class for Premium Passenger class.

Class:

Part 1: Flight classes

Class1: Flight (Base class)

- This is the Flight class used as the Base class in this method the Public attributes are flight number, origin, destination time, bookings.
- It created the encapsulation as it uses case price , availability economy, available business as private attributes and also it uses getter method for base price as encapsulation method.
- In this polymorphism also used in where for pricing calculations the overridden method as been used.

Class 2: DomesticFlight

- In this additional attributes has been created which is state tax (default 8.0 for 8%)

- This also inherits from Flight base class and calling parent constructor using super()
- Add state tax: price * (1 + state_tax/100) this is the formula for getting price value from base class and returns the final price.

Class 3: International Flight

- In this Boolean function as used for visa requirements.
- This class also used encapsulation for international fee as private attribute which is default price as $150.
- This also same like Domestic flight which inherits from Flight base class using super().

Part 2: Passenger Classes

Class 4: Passenger (Base class)

- The Attributes that used are passenger id, name, age, email
- They used polymorphism by using overridden method in get function in discount and Baggage allowance.
- In Discount they return 0 as standard and Baggage allowance they returned 20 kg as standard in get function.

Class 5: Adult Passenger

- In this it uses inheritance were it inherits the passenger class in adult passenger
- Then it pass the function.

Class 6: Child passenger

- In this class the new attributes guardian name
- This also using inheritance which it inherits the passenger class in child passenger
- Then it is using polymorphism by using get function and keep the discount default as 25%.

Class 7: Senior Passenger

- This is using inheritance which it inherits the passenger class in Senior passenger and These child, adult, senior passenger inherits from passenger that shows the multi-level inheritance.
- By using polymorphism of overridden method it uses in get function and made the baggage allowance as make default discount as 15%.

Part 3: Loyalty Program

Class 8: LoyaltyMember

- In this encapsulation are used which private attributes for miles in that three tier limits has been creates based on miles .
- Silver tier - upto 25000
- Gold tier - 25000 to 74,999
- Platinum tier - equal to 75000
- The discount for the each tier has been varies for example

  Silver: 5%, Gold: 10%, Platinum: 15%

Class 9: Premium Passenger

- This class where the Multiple inheritance has been used where it takes two base class or parent class Passenger and Loyalty Member.
- Using Get function using that it will give discount and baggage allowance from passenger class and by add inherits the Loyalty member class the bonus will be created these are the bonus Silver: +5kg, Gold: +10kg, Platinum: +15kg.

Part 4: Booking System

Class 10: Booking

- In this class the attributes that are used are Booking id where it generate with initial letter ”BK” + random 4- digit number. Other attributes that are used are Passenger, flight, Seat class.
- In this class encapsulation is used where total cost and calculate cost is the private attribute.
- For booking confirmation Boolean function as been used where it starts as False
- To Apply passenger discount this is the formula that is used price * (1 - passenger.get_discount()/100).
- For confirm booking: put an condition if available seats are their then add booking to flight's bookings list. If passenger is LoyaltyMember, add miles (used distance: Domestic=1000, International=3000).Prints confirmation message with total cost. If condition does not satisfy it will return False and print error.

- For cancel booking : If cancellation booking confirm it will release seat back to flight calculate the refund which is 75% of total cost . This cancel confirmation remove from flight's booking list. Then it prints cancellation details and it will return the refund amount. If the cancellation is not confirmed it will print error and return 0.
- Example of booking format and details:
    a) Booking BK8741: Sophie Martin on Flight UA500 Class: Economy | Cost: $787.50 | Status: PENDING
    b) Booking BK8741: Sophie Martin on Flight UA500 Class: Economy | Cost: $787.50 | Status: CONFIRMED

Conclusion:

To conclude with this project "Airline Booking System" is used multiple Object Oriented Programming(OOP) concept were Encapsulation, Polymorphism using overridden method and Inheritance were these principles in creating this project. By using these principle in Booking system helps in smooth implementation of classes. This will help to integrate the function and makes process and run time easier. This Python code is high level language were it is used to solve these complex problems and helps this project. The main benefits of this OOPs concept is code reusability, simple, data redundancy and Data security. This illustrates that how OOPS concept is used through Python language and makes it work.