* Project Goal:
  + Design and create a program that acts as a travel planner
* Characteristics of the Project:
  + Allows user to:
    - Choose the country of destination
    - Choose from 4 categories for their desired country:
      * Hotels
      * Food
      * Attractions and Activities
      * Airplane tickets
    - User can create a budget for their trip
    - They will be able to search through each of the categories based on:
      * City: 3 cities for each country
      * Cost
        + Hotels: Cost per night
        + Airplane: Roundtrip ticket price
        + Food and Attractions/Activities: number of dollar signs represents cost
      * Ratings
        + Only for hotels (out of 5 stars and can have half of a star)
      * Airlines and number of stops and airplane class (economy, business, first, premium)
    - System to compare trips
      * User can choose to create another trip to the same destination
      * User can then compare the total cost of each planned trip to see which one best meets their budget and interest
      * At the end once users choose their desired itinerary, the program will print out a summary of their trip
      * Properties:
        + Usage of relational operators
* Classes and behaviors of Project:
  + Behaviors:
    - Storing each trip as an object in class (Trip)
    - Storing a list of trip as a vector (list)
  + Classes:
    - Trip

|  |
| --- |
| Class: Trip |
| Objects/Data:  -string destination  -int NumOfDays  -Airline airplaneTicket  -Food restaurant  -ThingsToDo Activities  -Hotel sleeping  -Double EstCost |
| Behaviors:  +void CalcCost()  +double getEstCost()  +operator >  +operator <  +operator ==  +getAirline(): Airline  +getFood(): Food  +getActivities(): ThingToDo  +getHotel(): Hotel  +getDestination(): string  +getNumOfDays(): int  +compareAirlines(): void  +compareFood(): void  +compareAttractions(): void  +compareHotel(): void  +calcEstCost(): void |

* + - Airline

|  |
| --- |
| Class: Airline |
| Objects/Data:  -string airline  -string class  -int numberOfStops  -double cost |
| Behaviors:  +getAirline: string  +getClass: string  +getnumberOfStop: int  +getCost: double |

* + - Food

|  |
| --- |
| Class: Food |
| Objects/Data:  -string restaurant  -string cityName  -int numberOfStars  -string LevelOfCost |
| Behaviors:  +getRestaurant: string  +getCityName: string  +getnumberOfStars: int  +getLevelOfCost: string |

* + - Attractions

|  |
| --- |
| Class: ThingsToDo |
| Objects/Data:  -string AttrName  -string cityName  -string LevelOfCost |
| Behaviors:  +getAttractionName: string  +getCityName: string  +getLevelOfCost: string |

* Hotels

|  |
| --- |
| Class: Hotels |
| Objects/Data fields:  -string HotelName  -string cityName  -double cost |
| Behaviors:  +getName: string  +getcityName: string  +getCost: double |

* + - TripPlanner

|  |
| --- |
| Class: TripPlanner |
| Objects/Data:  -vector<Trip> List  -double EstTotalCost  -string nameOfUser  -double budget |
| Behaviors/Functions:  Constructor:  +TripPlanner(vector<Trip> list)  +addTrip(Trip newTrip): void  +deleteTrip(Trip findTrip): void  +CalcEstCost(): double  +withinBudget(): bool  +getBudget(): double  +getName(): string  +printTrips(): void  +operator >  +operator <  +operator ==  +sortTripCosts(): void  +printTrips(): void  +chooseTrip(): void  +printChoice(): void |

* System for estimating costs:
  + Airplanes:
    - Units: roundtrip ticket cost per person
    - Dependent on class
  + Restaurants:
    - Units: US dollars per person
    - $: 10 US dollars
    - $$: 20 US dollars
    - $$$: 40 US dollars
    - $$$$: 80 US dollars
  + Hotels:
    - Units: one person/one bedroom per night
  + Attractions/Activities:
    - Units: US dollars per person
    - Free: zero dollars
    - $: 5 dollars
    - $$: 10 dollars
    - $$$: (if any) 30 dollars