Project Documentation

Requirements Specification:

- All numeric user input must be positive: The program will have to check all user inputs for errors and address them accordingly.
- Max weight limit of plane cannot be exceeded: Program will have to not allow the user to load any cargos that exceed the weight limit.
- Max volume limit of plane cannot be exceeded: Program will have to not allow the user to load any cargos that exceed the volume limit.
- Duplicate cargo containers are permitted: Program should allow multiple cargos with the same label/name.
- Must ask user for data in a certain order. Program should request data from user in the following order 1.label 2.weight 3.height 4.width 5.length
- Continue to prompt user for data until specific conditions are met. Program should continue to request cargo data from user, unless user quits or until no more cargos can fit in the plane.
- Weight data must output in 2 decimal points.

Design Specification:

1. All numeric user input must be positive.
   To solve the problem all functions that request numeric data from the user or file will check and make sure that the value is positive and that it is not a character or string. If the user does not meet the conditions then the program should repeatedly request the data until the user enters a correct value.
2. Max weight limit of plane cannot be exceeded and Max volume limit of plane cannot be exceeded.
   To solve this problem, the program will have to update the used weight and used volume each time a new cargo is to be loaded, and check if the cargo that will be loaded does not exceed the weight and volume limit. If it exceeds the limit then the cargo will not be loaded and then the program will ask for a different cargo to be loaded instead that will not exceed the limit.
3. Duplicate cargo containers are permitted.
   To solve this problem, the program will use an array of cargo objects. The cargo object array each has their own location in the array. The location values will allow for multiple cargo objects with the same label because each will have their own location that can be used to identify each cargo object separately.
4. Continue to prompt user for data until specific conditions are met.
   To solve this program, the program will only end when the user enters quit for the label, and then the program will output the statistics and the table of cargos that was loaded.

To do this we put the main program functions into a loop that will only end when specific conditions are met.

Test Plan:

- What is to be tested?
  1. Cargo structure functions.
  2. Successful creation and initialization of Cargo array
  3. Loading the data from the external file into the array
  4. Printing out data in correct format/layout
  5. Functions returning correct values
  6. File contains correct data
  7. usedWeight and usedVolume is updated correctly
  8. maxWeight and maxVolume is not exceeded
  9. If allCargo file exist
  10. If data in allCargo file does not exceed the limits
  11. Cargos are added/loaded correctly
  12. Errors and exceptions are handled correctly
  13. All user inputs meet specific conditions
  14. Cargos that do not fit in plane are not loaded
  15. Functions are passing variables correctly

- Testing order within each type of test
  1. Cargo structure functions and variable members are accessible and contain correct values.
  2. If allCargo file is in correct format and can be loaded correctly
  3. If incorrect user inputs are not loaded
  4. If error/exception handlers are handling all types of errors and user input errors correctly
  5. If statistics and data are outputted correctly and in correct format/layout
- Assumptions made
  1. User will accidently enter negative values
  2. User will accidently enter characters into numerical values
  3. allCargo.txt file does not exist or cannot be loaded or file is in incorrect format/layout
  4. User will accidently enter more cargos then the cargo array size
  5. User will enter multiple cargos with same label and values
  6. User will enter values that will exceed limits
- Algorithms that may be used
  1. Search algorithms that will search the cargo array for the largest/heaviest cargos.

Test Cases and Results:

1. Cargo structure completed successfully with no errors the first time around.
   All variable members of Cargo are initialized correctly and all functions returns desired results.

2. Testing of loading allCargo file ended successfully. Program will check if file exist first, then it will check if the file is in correct format then it will check if the cargo values in the file does not exceed the max weight and max volume of the plane.
3. Testing of user input data ended successfully. Program will not allow negative and character values to be inputted into numerical variables, and will repeatedly ask user for the data until a valid value is entered. This test began with me entering large and random values for weights, height, length, width. To test if the program will prevent the cargo that exceed limit would not be loaded.
4. All error/exception handlers are handling errors correctly.
   To test error handlers I purposely created errors and tested if the errors was handled correctly.