CIS 22B Lab 2 Itty Bitty Airfreight (IBA)

#### 200 Points

Topics:

struct

class

struct to class

constructors

destructors

mutators

accessors

-----------------------------------------------------------------------------------------------------------------------------------------

Welcome to Itty Bitty Airfreight. We are a small, one airplane, local, airfreight company. We specialize in freight charters, special handling and quick turnaround shipments. If you need something delivered in the West, we are your solution. We do the smaller, local airports as well as the major airports.

What we need from you, the programmer is the following:

An object oriented, Unit Load Delivery (ULD), setup and tracking program. A Unit Load is one unit, either Container or Pallet to be loaded onto our Boeing 737. Here’s your data to setup and manage

Unit and type: Container or Pallet: AYF, AYK, AAA, AYY type Containers

and PAG, PMC, PLA type Pallets (a pallet is a flat platform for holding strapped down cargo).

Unit ID: Container or Pallet type: a five digit + airline code; our ID code is IB, e.g. AYF12345IB

Aircraft type: Ours is a 737; we are planning on adding more planes in the future of course.

Weight: The weight, in pounds, of the loaded container or pallet.

Destination: A three alpha character IATA string, e.g. MFR (Medford, OR), or PDX (Portland, OR)

Refer to the file airfreight.doc or airfreight.docx for sample struct or class information

### Lab 2.1 struct

Write a complete program with a struct to hold the data for one unit (container or pallet). Load the pertinent data and then output it to the screen. Use the following functions:

main

\* Uses new to obtain space for the data structure

\* Calls the other two functions

\* Deletes the space obtained using new

input

\* Reads all the data from the user

\* Puts all this data into the structure

output

\* Prints the data in a neat format

Put the main function first.

Use the function names and specified above.

Arrange the functions in the order listed above.

Use the following **data and field names**:

uld - Pallet

abbreviation - PMC

uldid - PMC46890IB

aircraft - 737

weight - 1289

destination – BFL

### Lab 2.2 struct to class

Utilizing the code in 2.1, convert your struct to a class with **private** data. Use new to allocate space on the heap for your object. You will need to provide complete object oriented code to support your program. That means a class, default constructor, constructor taking five arguments, mutators (setters), accessors (getters), a destructor plus input and output functions. Refer back to airfreight.doc or airfreight.docx for helpful information.

In main, create a load object on the heap using the default constructor. Output the contents of the object. Using your input function, set the data up for your object using the following information:

uld – Container

abbreviation - AYF

uldid – AYF97326IB

aircraft - 737

weight - 1710

destination – SMF

In the input function, **utilize mutators** to set up your data

Destroy your unit object using delete; put a cout to verify the deletion

Hints:

Pass your object from main by reference. This allows your input and output functions to work much

more easily. You will need this in later labs.

Your output should look like this (your output data is different, of course)

Unit load type: Pallet

Unit load abbreviation: PMG

Unit identifier: PMG12345IB

Aircraft type: 737

Unit weight: 1257

Destination code: SJC

Cargo destructor called