**Jacob Sawyer  
Student ID: 010397170**

**Submitted:**

**A: Algorithm identification:**

**B1: Logical Comments:**

I will go over comments made for *each* of the python files:

hash.py:

main.py:

package.py:

truck.py:

**B2: Development Environment:**

Hardware: Lenovo ThinkPad T480

A screenshot of a computer

Description automatically generated

Software:

Windows:

A screenshot of a phone

Description automatically generated

Visual Studio Code:

A screenshot of a computer code

Description automatically generated

Python:



**B3: Space Time and Big O:**

**B4: Scalability and Adaptability:**

**B5: Software Efficiency and Maintainability:**

**B6: Self Adjusting Data Structures:**

**C: Original Code:**

Code is fully a fully original implementation. Runs without errors.

**C1: Identification and Information:**

**A screenshot of a computer

Description automatically generated**

**C2: Process and Flow Comments:**

The program contains **two** main parts (this is identified within code as well):

1. Loading data from csv into HashMap -
2. Loading into trucks and delivering -

**D: Data Structures:**

The “bread and butter” of this program is the hash map. This contains all of the addresses and information needed to complete the deliveries. This is contained in its own python file labeled (hash.py). The class contains the following functions (*more details in section D1*):

**Init-** this initializes the HashMap. The initialization loads the HashMap, allowing the insert function to work.

**Insert**- this inserts the data into the HashMap.

**Search**- this looks for data in the HashMap. Can be used to locate a single value or multiple. Looks based on the key being the ID of the package.

**Delete-**

The reason this specific data structure was chosen is because of the usefulness of a key value pair. The packages we are given have a ton of variable info and some of it is the same amongst packages. However, all of the packages **do** have unique IDs which is perfect for a hashmap.

**D1: Explanation of Data Structures:**

**E: Hash-Table:**

**F: Look-Up Function:**

**G: Interface:**

**G1: First Status Check:**

**G2: Second Status Check:**

**G3: Third Status Check:**

**H: Screenshots of Code Execution:**

**I1: Strengths of Chosen Algo:**

**I2: Verification of Algo:**

**I3: Other Possible Algos:**

**I3A: Algo Differences:**

**J: Different Approach:**

**K1: Verification of Data Structure:**

**K1A: Efficiency:**

**K1B: Overhead:**

**K1C: Implications:**

**K2: Other Data Structures:**

**K2A: Data Structure Differences:**

**L: Sources:**

C950 Webinar-1 – Let’s go Hashing – Complete Python Code:

[Link to Repo](https://srm--c.vf.force.com/apex/CourseArticle?id=kA03x000000e1fuCAA&groupId=&searchTerm=&courseCode=C950&rtn=/apex/CommonsExpandedSearch)

Adopted code from this repo for hashing function.

**M: Professional Communication:**

Nothing to add. Thanks for reviewing my submission!