

CS/CE 1337 – PROJECT 5 – Interplanetary Delivery Service

Pseudocode Due:	CS/CE 1337.001 Mon/Wed class	10/30 by 11:59 PM in eLearning
	CS/CE 1337.002 Tue/Thu class	10/31 by 11:59 PM in eLearning
Project Due:	CS/CE 1337.001 Mon/Wed class	11/13 by 11:59 PM in eLearning
	CS/CE 1337.002 Tue/Thu class	11/14 by 11:59 PM in eLearning

Submission: All files are to be submitted in eLearning. Please submit the program's .cpp file for the project and either a Word document or PDF for the pseudocode. Projects can be submitted after the due date with a penalty of 3% per 2 hours late. Pseudocode will not be accepted late.

Problem: You have been hired by an interplanetary delivery service to keep track of shipments throughout the solar system. Shipments can be made between any two planets in the solar system (excluding Pluto since it lost planetary status recently). The shuttle which is used for transportation can never hold more than 10 Earth tons. Your program should record the shipping information in an output file.

Requirements: The program must use a structure for the planets and the shuttle. The planet structure must keep track of the following items:

- Planet name
- Distance from sun
- Surface gravity factor

Your program should create an array of planet structures initialized by reading data from a file. See the table at the end of this documentation for the information.

The shuttle structure should contain the following data:

- Cargo type
- Pick-up planet
- Delivery planet
- Speed (in miles per hour)
- Total distance to travel
- Weight in pick-up planet pounds
- Weight in delivery planet pounds
- Weight in Earth pounds
- Travel time for minutes
- Travel time for days
- Travel time for years

At a minimum, the program should use a function for each of the following tasks:

- Retrieving user input
- Calculate the number of hours necessary to make the delivery
- Convert the hours necessary to make the delivery to days and years

- Convert weight to Earth pounds
- Convert weight to other planet pounds

Input: Your program should validate all input. Do not assume the user will enter data of the proper data type. Continue asking the user for input until the user enters valid information. Ensure that your program will not crash if a string is entered for a numeric value and vice versa.

The user will enter the cargo type, pick-up planet, delivery planet, weight of the cargo in pick-up planet pounds and speed of the shuttle (maximum speed being the speed of light: 670616629 mph). You may use a menu to prompt the user for the pick-up and delivery planets, validating as necessary. Remember that the shuttle cannot hold more than 10 Earth tons of cargo.

The pick-up planet and delivery planet cannot be the same.

Output: The program should output the shipment information to a text file named Delivery_Report.txt. The shipment information should include the following:

- Cargo of the ship
- Weight of the cargo in Earth pounds
- Weight of the cargo on the other planets involved in the shipment
- The distance the shipment will travel
- The time it will take to deliver the cargo (based on the speed of the shuttle as entered by the user) in hours, days and years

Each shipment record should be separated in some way from other shipping records. Create your program in such a way that the output file does not get overwritten between executions of the program. The user should have the option to enter as many shipping records as he/she wants every time the program executes.

Planet Information:

Planet	Distance from Sun (Millions of Miles)	Surface Gravity Factor
Mercury	36	0.27
Venus	67	0.86
Earth	93	1.00
Mars	14	0.37
Jupiter	483	2.64
Saturn	886	1.17
Uranus	1782	0.92
Neptune	2793	1.44