

## Programming Assignment #2

### Small Store Inventory

CS 2308 Fall 2017

Instructor: Yijuan Lu

**Due: Wednesday, 10/04/2017:** upload electronic copy by 11:00am.

---

#### Problem:

Write a C++ program that will allow a user to manage the inventory of a small store that sells various products of any type.

The inventory for the store will contain the following information for each product:

|              |  |
|--------------|--|
| product name | (i.e. "AppleiPhone3GS", it will <b>NOT</b> contain spaces in it) |
| sku          | (stock keeping unit code, an integer)                            |
| quantity     | (how many of this product in stock)                              |
| price        | (in dollars and cents)   |

Note: Your program should be able to store up to 100 different products. You may assume that the skus will be unique (you do not need to check for this).

The program should first read the inventory from a text file named "inventory.dat". This file will contain data for each product in the inventory in this order: product name, sku, quantity, price. It will contain up to 100 products. See the sample file (inventory.dat) on the Tracs website.

Then, it should offer the user a menu with the following options:

1. Display the inventory sorted by sku.
2. Lookup a product by sku.
3. Lookup a product by name.
4. Quit

The program should perform the selected operation and then re-display the menu.

**Do not change the menu numbers** associated with the operations.

For the **Display** operation, display the information for each product on a separate line. The values should line up in columns (use setw). Headers for the table are optional.

For the **Lookup** operations, label the output values (i.e. Name: AppleiPhone3GS, etc.). If the product is not found, display an appropriate message.

### Additional Requirements:

- ☐ This program must be done in a **Linux or Unix** environment, using a command line compiler like g++. Do not use codeblocks, eclipse, or Xcode to compile.
- ☐ Your program **must compile** and run, otherwise you will receive a score of 0.
- ☐ The program must be **modular** (use top-down design), with significant work done by **functions**. Each function should perform a single, well-defined task.
- ☐ Use a **partially filled array** of structures to store the inventory:  
Use a counter variable to count the number of products that are read in from the file, and use this value as the size of the array for the search and sort functions.
- ☐ Your program should work for an input file with any number of products up to 100, but if you are struggling to make this work, you may assume it will have exactly 7 (for partial credit).
- ☐ You MUST use **binary search** for Lookup by sku.
- ☐ You may use (and modify) the code from the book. See the "Homework Resources" under Resources tool in TRACS. Please look at this code before you start implementing your program.
- ☐ I will put a sample input file on the Tracs website (inventory.dat) and the console output from running my solution on that file (output2.txt).

### Logistics:

Name your file **assign2\_XXXXX.cpp** where XXXXX is your TX State NetID (your txstate.edu email id).

There are two steps to the turn-in process:

1. Submit an electronic copy using the Assignments tool on the TRACS website for this class.
2. Submit a printout of the source file at the beginning of class, the day the assignment is due. Please **print your name on top of the front page**, and staple if there is more than one page.

See the assignment turn-in policy on the course syllabus (Tracs) for more details.