**Assignment #6**

**Introduction to C Programming – COP 3223**

**Objectives**

1. To reinforce knowledge of previous programming constructs
2. To learn how to use arrays

**Introduction: Ultimate Computing Fun**

A new game store is opening in your area! Ultimate Computing Fun (UCF) Games is looking for a home for their wide variety of computer games and has decided to make you a part of their growing business.

UCF Games has been up and running for a few weeks now and has been very successful. They would now like your assistance in creating an inventory tracker to keep up with sales.

**Problem: Inventory Tracker (tracker.c)**

The goal of the UCF Games inventory tracker is to maintain a record of the current inventory, update the inventory as games are delivered and purchased, and alert the user when new games of a specific type should be ordered.

This program will read from a file to initialize the program with the current stock for 5 different genres of games. While strings could be included, these genres can be labeled 1-5 for now.

After the auction information is initialized, your program should give the user the following options.

1. Increase Inventory
2. Decrease Inventory
3. View Inventory
4. Quit

If the user enters 1 or 2 you should ask which genre of games they want to increase/decrease respectively. The user will then enter a number 1-5 to indicate their desired genre. Ask how much they wish to increase/decrease stock. For increases, you should increase the amount of stock for that genre.

For decreases, you should first check that the decrease will not take the inventory below zero. If the decrease is valid, update the amount of stock for that genre. If the decrease is not valid, tell the user they do not have enough stock to decrease that genre.

After a decrease in inventory, check for any “low-stock” warnings. That is, if any stock is below 100, print a message that says “You should order more of genre X” where X is the number for that genre.

If the user chooses to view the inventory, print out a table of the inventory as follows:

Genre Stock

1 A

2 B

3 C

4 D

5 E

Where A, B, C, D, and E are integers that represent the number of games in each genre.

Finally, reprint the menu to see if there is another task the user wishes to perform.

If the user selects “quit” there should be no further printouts and the program should terminate.

If the user enters an invalid input at any time, you should re-prompt the user for the correct input.

**Input Specification**

1. All inputs will be integers greater than or equal to zero

**Input File Specification**

The file will contain 5 integers specifying the starting amounts for each genre.

**Input File Sample**

1500 400 650 743 854

**Output Sample**

Below is a sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity’s sake.)

**Sample Run**

**Welcome to the UCF Games Inventory Tracker!**

**What would you like to do?**

**1. Increase Inventory**

**2. Decrease Inventory**

**3. View Inventory**

**4. Quit**

*3*

**Genre Stock**

**1 1500**

**2 400**

**3 650**

**4 743**

**5 854**

**What would you like to do?**

**1. Increase Inventory**

**2. Decrease Inventory**

**3. View Inventory**

**4. Quit**

*1*

**Which inventory would you like to increase?**

*3*

**How much stock would you like to add?**

*250*

**What would you like to do?**

**1. Increase Inventory**

**2. Decrease Inventory**

**3. View Inventory**

**4. Quit**

*2*

**Which inventory would you like to decrease?**

*2*

**How much stock would you like to remove?**

*350*

**You should purchase more games for genre 2.**

**What would you like to do?**

**1. Increase Inventory**

**2. Decrease Inventory**

**3. View Inventory**

**4. Quit**

*3*

**Genre Stock**

**1 1500**

**2 50**

**3 900**

**4 743**

**5 854**

**What would you like to do?**

**1. Increase Inventory**

**2. Decrease Inventory**

**3. View Inventory**

**4. Quit**

*6*

**Sorry, that was not a valid input.**

**What would you like to do?**

**1. Increase Inventory**

**2. Decrease Inventory**

**3. View Inventory**

**4. Quit**

*5*

**Deliverables**

One source files – *tracker.c* – is to be submitted over WebCourses.

**Restrictions**

Although you may use other compilers, your program must compile and run using Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

**Grading Details**

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility – You must submit C source files that can be compiled and executed in a standard C Development Environment. If your program does not compile, you will get a sizable deduction from your grade.