# Comp 150 – Assignment Two

***Due: see Important Dates at the Comp150 portal on myclass.ufv.ca***

You will write some form of record management system. Some ideas of programs that manage records are:

* Address Book
* Media Collection
* Stamp Collection
* Actually any form of Inventory Control will suffice; bring me your idea for approval BEFORE starting on the Design Draft.

The SimpleInventory Program (found with the assignment documents) shows how a large part of the work is completed. DO NOT just copy it. Use is as a resource in writing your own specialized program. We will also cover it as part of Unit 10.

The following requirements must be met to ensure full marks (this is in addition to requirement found in the documents listed at the end of this document):

* Fully modularize your program
* Use pass by reference parameters wherever appropriate
* You must use record (structs) stored in an array or vector
* Your program must be able to save the vector of input data to a file and reopen that data file later (you MUST include a data file in your submission).
* Your program must be menu driven
* The menu shall have following functionality:
  + Enter a single new record (same as example)
  + Enter many records one-after-the-other option (not in the example)
  + Edit and Delete records
  + Search for and display one record in detail (part of this functionality is required for edit and delete suggesting that “search” is a function used by more than one process)
  + Display a listing of all records (truncate appropriately so that all important information appears on one line)
  + Sort by name or id of record
  + Sort by one other field of your choice (you only need two sorts in total)
  + Save to file
  + Open file (the file that you include must be populated with detailed and extensive data of no less that 5 records; summary display must truncate longer fields when displaying all records while detail display shows all data for one record)
  + Quit

**IMPORTANT**: do not allow a process to attempt to work on an empty container; display a message that there is no data to process or display

Here is a list of the things that I will look for in determining your grade:

* Input from the keyboard and output to the monitor (style, ease of use, ease of understanding)
* Menu system
* Variables and constants
* Containers (array and vector)
* File I/O
* Sorting
* Logic for selection and iteration
* Modularization
* With high cohesion and low coupling
* Using both pass by value and pass by reference (don’t use pass-by-ref where a return statement will do)
* You have included a data file having a minimum of 5 records
* ABOVE ALL, I will be looking to see that you are using appropriately applied structured programming concepts and methodologies from the entire course

## RULES FOR COMPLETING THE ASSIGNMENT

* The code you submit MUST compile and run without error in order to qualify for grading. I do NOT mark code that does not compile and run.
* You MUST fully document your code per the commenting document.
* Your code MUST be clear, easily readable and understandable. Some of the issues include good choices for symbol naming, indentation standards and consistent bracing. Additionally do NOT exceed about 60 characters of line width.
* The code MUST demonstrate all concepts as documented in the lecture outlines (as appropriate).
* You MUST include instructions for running your program and appropriate data to enter during the run. Place these in the code comments under “Run Instructions” at the beginning of the file (see Comp150&152\_Commenting.v02.docx). Note that I will still enter inappropriate values to test robustness.
* Prompts MUST clearly indicate what has to be entered. Test this by letting someone else (who is completely unfamiliar with your work) try to run your program. This is also a good way to uncover problems in your design. Let them tell you just how bad it is then fix it!!!
* You MUST include a working data file that has been populated with appropriate data and consist of at least five records.

## RULES FOR THE SUBMISSION PROCESS

* Submit only your final version and your data file
* Rename your final version (depending on your section) :
  + lastnamefirstnameA2Ab1.cpp OR
  + lastnamefirstnameA2Ch1.cpp
* Zip up (RAR is not acceptable) and email your completed code to me at [paul.kroeker@ufv.ca](mailto:paul.kroeker@ufv.ca). If you wish to write me a note DO NOT put the note into the email body; include it the opening comments of your code file. Name the zip file (depending on your section) :
  + lastnamefirstnameA2Ab1.zip OR
  + lastnamefirstnameA2Ch1.zip

## OTHER FILES TO REVIEW

* Comp150&152\_Commenting.v02.docx
* SimpleInventory.cpp
* Inventory.txt (this is the data file for the Simple Inventory program)

END