

COURSEWORK ASSIGNMENT

UNIVERSITY OF EAST ANGLIA

School of Computing Sciences

Unit: CMP-5015Y
Assignment Title: Assignment 2 – C Programming (PROVISIONAL)

Date Set : Monday, week 4 (16/10/17)
Date & Time of Submission : Wednesday, week 8 (15/11/17 3 p.m.)
Return Date : Wednesday, week 12 (13/12/17)
Assignment Value : 10%
Set By : Dr G. C. Cawley **Signed:**
Checked By : Dr A. Bagnall **Signed:**

Aim:

The aim of this assignment is for the student to gain experience in the design and implementation of a relatively small program in the C programming language. This will involve the implementation of abstract datatypes, using the `struct` - the forerunner of the `class` in C++ and Java, providing insight into the historical development of programming languages. Understanding the common features of programming languages makes it easier to learn new programming languages in future, a key skill for programmers.

Learning outcomes:

On successful completion of this exercise, the student will have reinforced a basic understanding of the concepts of abstract data types (ADTs) and their use in writing maintainable modular code, and will be familiar with the syntax of the C programming constructs, including file I/O.

Assessment criteria:

Marking Scheme (out of 100 marks):

Marks will be awarded according to the proportion of specifications successfully implemented, programming style (indentation, good choice of identifiers, commenting etc.), and appropriate use of programming constructs. Marks may also be awarded for correct use of more advanced programming constructs not covered in the lectures. Program for maintainability – ensure that your programs are easily understood by a human reader, as well as correctly followed by the computer; it is not sufficient for the program to merely run, generating the correct output.

Description of assignment:

Chartlins is a company that specialises in providing electronic components for hobbyists to build simple electronic devices of their own design, or from a kit of parts. They have hired you to write a bespoke system to manage their inventory of components and process sales etc. The program must be written in the C programming language (conforming to the C11 standard) and demonstrate the use of `structs` and data abstraction to write modular, maintainable programs (as the maintenance of the system will be carried out by in-house software support technicians). The file `inventory.txt` provides details of the companies initial inventory of components (resistors, capacitors, diodes, transistors and integrated circuits). For each stock item, the file provides the component type, the stock code, the number of items in stock and the unit price in pence. The remainder of the record provides some additional information:

- Resistors – the resistance in Ohms, encoded according to the BS1852 standard (see http://www.electronics-tutorials.ws/resistor/res_2.html for further details).
- Capacitors – the capacitance in Farads, encoded in a manner similar to resistance.
- Transistors – an indication whether they are NPN, PNP or FET devices.
- Integrated Circuits (ICs) – a brief description.

The file `sales.txt` contains a list of sales transactions in which customers have bought one or more components from the company. The file describes the date of the transaction, the stock code of the component purchased and the number of items purchased. Your program should apply the transactions described, in this file, generating appropriate error messages if a transaction cannot be completed. After the transactions have been performed, your program should determine the answers to the following sales queries:

- Print a list of the inventory, sorted in order of increasing price.
- Which day resulted in the greatest sales volume (give the day, the total number of items sold and the total amount of money spent (in pounds and pence)).
- How many NPN transistors does Chartlins have in stock after processing all successful sales transactions.
- What is the total resistance of all the remaining resistors in stock?

The program should consist of five files, which define the basis of the marking scheme:

- `StockItem.h` and `StockItem.c` : These files should define a `struct` that represents a stock item and a set of functions used to access the `struct` (forming an abstract data type) (30 marks).
- `Inventory.h` and `Inventory.c` : These files should define a `struct` representing the inventory of the company as a linked list (or other suitable data structure) and a set of functions used to access the `struct` (20 marks).
- `Sales.h` and `Sales.c` : These files should define a `struct` representing the sales data, as a linked list (or other suitable data structure), where each node in the link contains a reference to the `StockItem` structure, the date (you may use the Date ADT developed in the lectures) and the number of items sold. The list should only record the `successful` transactions (20 marks).
- `StockProgram.c` : This file contains the `main` part of the program that loads and processes the inventory and sales data, and provides answers to the four queries (10 marks).
- The remaining 20 marks are awarded for the answers to the four inventory questions listed above.

All questions regarding the specifications must be asked via the appropriate discussion board on the module volume on BlackBoard. Note that `inventory.txt` and `sales.txt` may change before the due date for the assignment, and the answers to the queries must be based on the *final* versions of these files.

Required:

A printed copy of your solution must be submitted to the hub, containing the printouts of your source code followed by a sample of the output from the programs (make sure your program does not generate an excessive amount of diagnostic information!), giving answers to the queries listed above. The files must be printed in the following order, starting `StockItem.h`, `StockItem.c`, `Inventory.h`, `Inventory.c`, `Sales.h`, `Sales.c`, `StockProgram.c`, followed by the output of C program giving answers to the queries. The code must be formatted so that it prints out legibly using a sensible font size (e.g. format code to be no more than 80 columns wide) and stapled securely in the top left-hand corner. Submissions where these instructions are not followed *will* be penalized.

An electronic submission *must* also be made via BlackBoard, using the PASS system. The code must be compilable using `gcc` (the compiler used by NetBeans). If you develop your solution on another platform, you are required to ensure that it compiles and operates correctly using PASS - do not leave this to the last moment, C is not as portable as Java and this may take some time. Submissions that do not operate correctly using the PASS system *will* be penalized.

Handing in procedure:

A printed copy of your submission must be submitted to the Teaching Hub, by the date indicated above. A .pdf file, generated using the PASS system, must also be submitted via BlackBoard.

Plagiarism:

Plagiarism is the copying or close paraphrasing of published or unpublished work, including the work of another student without the use of quotation marks and due acknowledgement. Plagiarism is regarded as a serious offence by the University and all cases will be reported to the Board of Examiners. Work that contains even small fragment of plagiarised material will be penalised.