Overview

The C programming language is considered a lower level procedural language meaning that it makes use of statements to change the state of the program and compiled, making it a highly memory efficient language. It does however mean that it must follow explicitly sequence of commands e.g. we cannot refer to a method or function unless it has been created; it solves the given problem in a top-down manner.

Alternatively, Java is an object orientated high level language. It utilises objects as templates, indicating the sorts of states the object can have and the functionality it can perform. Java performs automatic memory management so is not as efficient as C. Unlike C the execution method is interpreted which means it produces a result from the program which is different to C as this type produces a program written in assembly language [4].

Cerate Mappings

The C programming language uses structs to as a way of grouping individual variables together. A struct is a type of data, similar to an integer and is defined by the programmer and made up of one or multiple data types. Class is a value type and its object is created on the stack memory [2].

In our shop we have created 4 different structures to represent the Product, the Product stock, the Shop and lastly the Customer. If we look at the Customer struct we have indicated that we want there to be a name, and as we do not yet know how big that name will be we user a pointer(\*) which is used to allocate memory dynamically, the name will be a character (char) e.g. letter, number or symbol. The customer will also have some money to spend represented by the budget, the double indicates tells the complier that this will be a numerical value [3]. We also reference the productStock struct using a shoppingList array.

Java utilises objects as opposed to structs in C. Firstly in our Java program we created a Java Project. Java uses special methods called constructors to initialise objects, which is different to the structs in C which only store data. If we look at the Customer class, we have included a name which will be a String (characters), it will be private, meaning the method is private, or nobody but the Customer instance can access it. Next is the budget or cash the customer has to spend represented by a double or a number, this is similar to C whose syntax is also a double. Again similarly to C, Java uses an array List to create a shoppingList from the ProductStock object.

Reading in the CSV files

In Java, we create an entry point to the program using the *public static void* method and pass in the path to the files that are in the base of the project. We tell Java what each index in the array will be, we have said that line (0) will be different to the rest and to be treated as such. We then assign array [0] as a string as this is the product name, arr[1] is a double as there is decimal points and arr[2] is an integer as there are no floating points, this is important as the program will want to know what to expect from the csv file and how it should be treated.

To read in a file in C, we created a method that takes no arguments. Or each line we create a product stock variable and add it to the struct representing the shop. We tell the file to read the length of each line until it reaches the end of the file. To extract the different pieces of information we use the string tokenizer or strtok. Compared to Java this is quite low level. We make use of the pointers to pull each variable. We also need to convert each variable into the appropriate data types using atoi for integers and atof for floating point.

Printing Output

We create print methods in C, which takes in the shop struct for instance. Printf is the function used to pull together the information from the shop struct. We reused the printProduct method to pull in the name, price and quantity. A significant difference between C and Java is evident in this section of code, as C is a low level language it was necessary to dynamically allocate memory. To do this we made use of the string copy function, this made a copy of the string and placed it in the new memory location.

References:

[1] <https://www.techopedia.com/definition/3933/low-level-language>

[2] <http://net-informations.com/faq/oops/struct.htm>

[3] [https://www.thoughtco.com › definition-of-double-958065](https://www.thoughtco.com/definition-of-double-958065)

[4] <https://kb.iu.edu/d/agsz>