Multi-Paradigm Project Report

In my project I read-in from and udated csv files containing shop stock and customer order information in each of th e three implementations. There is also an option to become a live customer in which the order is got from real-time u ser-input rather than from a csv file. A display menu appears once the programs are run, so the user can select one of the options that are available. The options available are from numbers one to three; one is to display the shop conten ts, two is to update each customer csv in order and update the shop cash based on the money received and three give s the user the "live customer" option. If the user fails to enter correctly, they are returned to the menu and asked for i nput again.

In my object oriented python program, I created classes of Product, ProductStock, Customer, Shop and live_cus tomer. An object oriented program can untilize four different qualities, namely - polymorphism, encapsulation, inher itance and abstraction. "Polymorphism is an object-oriented programming concept that refers to the ability of a varia ble, function, or object to take on multiple forms. In a programming language exhibiting polymorphism, class object s belonging to the same hierarchical tree (inherited from a common parent class) may have functions with the same n ame, but with different behaviors." Encapsulation binds attributes and methods together into object and limits direct access. Encapsulation can also be used to hide information. When attributes or methods are bound to constructs inste ad of being bound to global variables, it allows a programmer to restrict access. Abstraction is when a user hides info rmation about an object, such as a class or method, in order to reduce the complexity and the lines of code in the pro gram. Inheritance may be defined as "deriving new classes (sub classes) from existing ones such as super class or ba se class and then forming them into a hierarchy of classes."

In my Python object oriented program, I created classes of Product, ProductStock, Customer, Shop and live_cus tomer. An object oriented program can untilize four different qualities, namely - polymorphism, encapsulation, inher itance and abstraction. Encapulation in the created class objects captures defining attributes. There is both an array c ontaining the shop stock and an array containing the customer orders. These arrays are filled when an instance of the 'Shop' or 'Customer' classes is called. After the shop stock array is filled with the products and information about the products, the program goes first to the '__repr__' method of Product, and then to the '__repr__' method of ProductSt ock. This is due to the shop stock array gaining functionality from the Product and ProductStock classes. Products fr om the shop csv file are added to the Product class, which has the attributes 'name' and 'price'. These products contai ning the name of the products and their prices are then added to the ProductStock class, along with the quantites got from the shop csv file. By selecting option one, the user can view this information. An instance of the 'Shop' class is created, with the relevant shop csv file passed in as an argument. Then using '.stock' on that instance and wrapping a print statement around this; the program goes to the '__repr__' methods and the information is printed. The '__repr__ ' method returns a string representation of an object, it is quite similar to the '__str__' method, a difference being, tha t the '__repr__' method can be used to create an object again. In contrast, in the Python procedural program, a separat e 'print_shop()' function was created to print out the shop information, with 'shop' passed in as the only argument. By returning the shop array, ie. 'shop', and not merely printing it out in the 'create_stock_shop()' function, it was possibl e to gain the functionality of this shop array in another function. The 'self' keyword is not used in the procedural prog rams, only in the object oriented program. It is used to create attributes of a class object inside the constructer 'init' m ethod. Using the self keyword joins these attributes to the class, so that elsewhere in the program the attributes will st ill be part of a particular class when that class is called somewhere else in the program.

As Python is an interpreted language is takes longer to execute than C programs. In C, the type of a variable mu st be declared when it is created, it only accepts values of that type later in the program. Whereas is Python, variable s are untyped. This means any one variable can be of a different type at different locations in a Python program or sc ript. Debugging in C is more difficult because it accesses the whole program at once, compiles it and then throws out any errors. C has a higher degree of complexity due to the complicated nature of its syntax compared to Python. Me mory management has to be undertaken by the user or programmer in C. C has much fewer built-in functions and lib raries compared to Python which has thousands. The C program is similar in some ways, it operates in a procedural manner, and has some of the same Python keywords such as 'while' and 'if', and more importantly, they work in the s ame way. Many keywords are similar such as 'printf'. The C language program requires the type of a variable to be d eclared when creating a variable, unlike in Python. When reading information from the shop csv file, it is required to change the type of some of the information, because when the 'strtok()' method is used to parse the file, the informat ion is outputted as string types. Also, memory space must be manually created by the user, using the 'malloc()' meth od unlike in the Python programs - Python does this automatically. In C, structs are created to store information and instances of the structs can be created later in the program. The asterisk inside a struct allows a 'pointer' to be created , this facility is not available in the Python procedural or objected oriented programs. The pointer is useful when iter

ating through a struct, such as the 'shop' struct. The index keyword can be used to increment by one or some other a mount, this can iterate through all the relevant information until the end of an array is reached, for example. The aste risk is used when parsing a csv file when the user doesn't know how much information, or how many lines the file co ntains. The procedural c and Python progams are considered top to bottom in the way the program runs them, where as the object oriented progam is said to be bottom to top.

Tuple-unpacking was carried out in the procedural Python program, inside for-loops; in order to extract the re quired information from the arrays inside the customer dictionary. This took up more lines of code than the object or iented approach, which is a common trait in a procedural program. This is a negative aspect of using a procedural ap proach. Reading many lines of code can prove cumbersome. A positive aspect of the procedural program, is the lowe r complexity and that it is easier to understand how a procedural program runs. When an array or a dictionary is retur ned, in a function, the array can be passed in as an argument into another function, this is similar to inheritance in an object oriented program, where classes can inherit from another class, or from multiple classes.